



PONTIFICIA  
UNIVERSIDAD  
CATÓLICA DE  
VALPARAÍSO



**Elim Manuel Vargas Villarroel**

# Implementación de un sistema de detección de personas en RPI

Informe Proyecto de Título de Ingeniero Electrónico



**Escuela de Ingeniería Eléctrica  
Facultad de Ingeniería**

Valparaíso, 24 de mayo de 2018



# Implementación de un sistema de detección de personas en RPI

Elim Manuel Vargas Villarroel

Informe Final para optar al título de Ingeniero Electrónico,  
aprobada por la comisión de la  
Escuela de Ingeniería Eléctrica de la  
Facultad de Ingeniería de la  
Pontificia Universidad Católica de Valparaíso  
conformada por

Sr. Gabriel Hermosilla Vigneau  
Profesor Guía

Sr. Francisco Pizarro Torres  
Segundo Revisor

Sr. Sebastián Fingerhuth Massmann  
Secretario Académico

Valparaíso, 24 de mayo de 2018

*Dedicado a:*

*Mi abuelo materno:*

*Juan Villarroel Bahamondes, (08/03/1921 – 01/04/2004)*

*Mi abuela materna:*

*Hilda Aguilera Espinoza, (17-02-1935 – 26/04/2004)*

*Mi abuelo paterno:*

*Víctor Vargas Vargas, (18-12-1927 – 07/01/2014)*

# Agradecimientos

Quisiera agradecer en primer lugar a mis padres, José Vargas y Sofía Villarroel, por su incondicional apoyo en estos años de estudio, por mantener la esperanza en mí hasta en los momentos más complicados y por ser quienes gracias a su esfuerzo, permitieron alcanzar mis objetivos. También agradecer a mis hermanos Nicolás Vargas y Francis Villarroel, quienes me aconsejaron y apoyaron desde un principio, siendo un pilar fundamental en los primeros años de universidad, en donde las cosas no son como uno las imagina.

Por otra parte, estas palabras de agradecimiento también están dirigidas a Valeria Torres, quien ha confiado en mí en todo momento, además de mantenerme con ánimo en todo el periodo de fabricación, desarrollo y exposición del proyecto de título, en donde más de una vez pensé que ya no había nada más por hacer.

Agradecer también a mis profesores Gabriel Hermosilla y Francisco Pizarro, quienes depositaron su confianza en mí para la realización de este proyecto, y los cuales siempre me aconsejaron de la mejor manera posible para alcanzar los objetivos propuestos, consiguiendo finalmente, un trabajo con el cual puedan sentirse realizados.

Un pequeño reconocimiento también a mis dos fieles compañeros que siempre estuvieron conmigo en las largas noches de estudios y de redacción de informes de proyecto, Moe y Milo, mis dos gatos.

Finalmente agradecer a todas las personas que de una u otra forma me permitieron lograr este objetivo, mis compañeros y amigos de generación, de carrera y de universidad, a los profesores y a la Escuela de Ingeniería Eléctrica PUCV.

*Valparaíso, 28 de mayo de 2018*

Elim Vargas

# Resumen

En el siguiente trabajo de tesis se propone desarrollar de un sistema cuyo objetivo es el de realizar detección de personas en distintas situaciones específicas, ya sea en video o en tiempo real. Además se desea localizar a estas personas señalando su posición en la escena ya su vez, realizar un conteo de las mismas.

Para la detección de personas se propone utilizar procesamiento digital de imágenes y visión por computador, por lo tanto se requiere de alguna librería que permita la utilización de algún descriptor en específico, y en donde se puedan realizar los procesamientos necesarios en las imágenes obtenidas.

Para llevar a cabo la detección de personas se optó por la utilización del descriptor HOG implementado en un computador de placa reducida, ya que son placas que cumplen la misma función que un computador, pero con una menor cantidad de recursos necesarios para su funcionamiento. En este caso, se decidió trabajar con una Raspberry Pi 3 debido a sus características, sin embargo, en caso de ser necesario se deben buscar alternativas que permitan un funcionamiento óptimo del sistema propuesto.

Para implementar un sistema que sea capaz de detectar y contar personas, es necesario desarrollar una serie de pruebas tanto en imágenes, videos y en tiempo real, para ello se requiere analizar situaciones con flujo de personas de manera controlada, para conocer el funcionamiento del detector. Luego corresponde someter al sistema a situaciones donde la el flujo de personas sea de manera no controlada, con el objetivo de poder implementar el detector en situaciones reales.

Las principales conclusiones de este trabajo hacen referencia a: (i) es posible realizar detecciones de personas en una RPI 3 mediante el descriptor HOG en tiempo real, por lo que se cumple con la metodología propuesta para el desarrollo del proyecto, (ii) el sistema logra como mejor resultado cerca de un 85% de efectividad en la detección de personas, (iii) existe una relación directa entre el lugar donde se realizan las detecciones, la altura de cámara y las características de la misma, con la detección correcta de personas.

Palabras claves: *Falsos Negativos, Falsos Positivos, Frame, HOG, RPI 3, SVM.*

# Abstract

The next work thesis is going to develop a system that's goal is to make a detection of people in specific situations, either video or real time. It is wanted to locate those people by showing their position in the image as well as counting them.

In order to detect people, it is suggested to use digital imaging processing and computer vision. Therefore, it requires a library that allows the use of a specific descriptor so the necessary processing in the images can be obtained.

The HOG descriptor was chosen to detect people, working in a computer which uses a reduced amount of necessary resources. This project was made in a Raspberry 3 Pi due to its characteristics, however, if necessary alternative options were sought out to allow the optimal functioning of the proposed system.

To develop a system that is able to detect and count people it is necessary to develop many tests with images, video and real-time for it is required to analyze cases where the number of people can be controlled to know the operation of the detector. Then, it is necessary to subject the system to cases where people's flow cannot be controlled, with the objective to implement the detector in real situations.

The main conclusions of this work refer to: (i) It is possible to carry out detections of people in an RPI 3 using the HOG descriptor in real time, (ii) the system achieves as the best result an 85% efficiency detecting people, (iii) there is a direct connection between the place where the detections were made, the camera's elevation and the characteristics of it, with de correct detection of people.

Key words: *False Negatives, False Positives, Frame, HOG, RPI 3, SVM.*

# Índice general

Introducción.....	1
Objetivo general.....	3
Objetivos específicos .....	3
1 Detección de personas .....	4
1.1 Procesamiento digital de imágenes.....	4
1.2 Visión por computador .....	5
1.3 Detección de objetos .....	6
1.3.1 Detección de personas.....	7
1.4 Problemática .....	8
1.5 Solución .....	9
1.5.1 Sistema desarrollado .....	9
1.5.2 Computadores en placas reducidas.....	9
1.5.3 Librería OpenCV .....	10
2 Descriptor HOG .....	11
2.1 Descriptor de características.....	11
2.2 Histograma en una imagen.....	12
2.3 Detección de bordes .....	13
2.4 Gradiente .....	13
2.5 Descriptor HOG .....	15
2.5.1 Construcción de HOG.....	17
2.5.2 Esquema de la construcción del descriptor.....	18
2.6 Máquina de vector de soportes.....	19
2.6.1 SVM para clasificación binaria para ejemplos linealmente separables .....	19
2.6.2 SVM para clasificación binaria de ejemplos cuasi-separables linealmente .....	20
2.6.3 SVM para clasificación binaria de ejemplo no separables linealmente .....	21
3 Raspberry Pi.....	23
3.1 Introducción a la RPI .....	23
3.2 Características generales.....	24
3.3 Lenguaje de programación en RPI .....	25

---

3.4 Aplicaciones realizadas .....	25
3.5 Modelos de RPI .....	26
<b>4 Construcción del sistema.....</b>	<b>28</b>
4.1 Trabajo previo .....	28
4.2 Estructura del descriptor .....	29
4.2.1 Estructura del sistema.....	30
4.3 Análisis de HOG en RPI 3 y Computador .....	31
4.3.1 Experimentación en imágenes .....	31
4.3.2 Experimentación en video .....	32
4.3.3 Experimentación en tiempo real.....	33
4.4 Análisis de lo realizado .....	34
<b>5 Experimentación.....</b>	<b>36</b>
5.1 Ajustes de parámetros extrínsecos .....	36
5.2 Ajuste de resolución de video .....	37
5.3 Ajustes de posición de la cámara.....	38
5.4 Experimento con flujo de personas controladas.....	39
5.5 Experimento con flujo de personas no controlado .....	40
5.6 Análisis de los resultados.....	42
<b>Discusión y conclusiones.....</b>	<b>45</b>
<b>Bibliografía .....</b>	<b>48</b>

# Introducción

El procesamiento digital de imágenes es un campo de investigación que se mantiene en constante desarrollo [1]. El continuo progreso en esta área se debe a la profunda relación con aplicaciones para la interpretación humana, como también para percepciones autónomas de máquinas [2], por lo tanto, son cada vez más los campos en donde se puede utilizar este tipo de desarrollo, por ejemplo en medicina, telecomunicaciones, inspección industrial, efectos artísticos, entre otros [3].

Uno de los campos en donde la visión por computador y el procesamiento digital de imágenes están tomando cada vez más relevancia, tiene relación con el ámbito de la seguridad, en particular el seguimiento de personas, por lo que buscar soluciones sencillas y económicas, además de efectivas, puede significar un gran avance tecnológico en este tipo de problemas. Sin embargo, lo que se presenta hoy en día es que muchos de los datos que pueden obtenerse con este tipo de procesamiento, no logran ser utilizados de buena manera, provocando que este tipo de sistemas no puedan ser masificados en el mercado [4].

Alguna de las aplicaciones en que se utiliza el procesamiento digital de imágenes, tiene relación con la detección de objetos para un posterior análisis. Para ello es necesario identificar aspectos propios del objeto en cuestión. De este modo, es como también se puede utilizar este tipo de metodología para la detección de personas, en donde existen diversas maneras para realizar detecciones dependiendo de las especificaciones que se deseen obtener, ya sea para solamente determinar la presencia o no de personas, como una posible identificación de estas.

Con el paso de los años el procesamiento digital de imágenes ha logrado avanzar en cuanto a las distintas áreas en donde pueda ser utilizada, ya que va de la mano con el avance tecnológico presente en los computadores. De este modo es como también es posible implementar diversas aplicaciones en computadores de placa reducida, donde existen diversas alternativas para el uso específico para el que se desea utilizar. Sin embargo, pueden presentar algunas limitaciones en cuanto a su funcionamiento y rendimiento en comparación a un computador convencional, esto debido principalmente a que no poseen las mismas características técnicas. No obstante, se ha decidido trabajar con una Raspberry Pi 3 para el desarrollo de un sistema que permita la detección de personas.

La Raspberry es en estos días uno de los productos más populares para el diseño de proyectos electrónicos debido a su bajo costo y la vez la gran cantidad de beneficios que trae consigo, ya sea por sus dimensiones físicas, como también por la conectividad que mejora con las nuevas versiones lanzadas al mercado, además es ideal para ser utilizado como un pequeño computador dedicado a realizar trabajos en específico. Cabe señalar que en sus inicios, esta placa había sido desarrollada para fines pedagógicos en niños, pero su masificación fue tan grande debido a las distintas aplicaciones a las que se les puede dar uso. Su principal ventaja tiene relación con su fácil manejo, ya que cualquier persona interesada en aprender a usarla puede hacerlo, además es muy fácil de adquirirla, puesto que desde internet hay un sin número de distribuidoras [5]. Es así como su implementación para distintos proyectos tanto de uso personal como a nivel industrial han resultado bastante favorables, permitiendo que empresas decidan utilizar este tipo de placas para la realización de sus proyectos, junto con su bajo costo en relación a otros productos con similares características [6].

Para implementar el sistema capaz de detectar personas, es necesario realizar una serie de pruebas, en distintas situaciones cada una de ellas. El objetivo es obtener la mayor cantidad de información en cuanto a elementos que afectan de forma directa o indirecta la detección de personas, ya sea por estructura del programa o por las características de la RPI. A su vez se realizará una comparación entre un computador de escritorio y la RPI 3, con el objetivo de determinar si es factible realizar este tipo de implementaciones en el dispositivos, debiendo someter el detector de personas a análisis tanto en imágenes fotográficas, videos y en situaciones en tiempo real.

Por otro lado es necesario realizar un detallado análisis sobre todos experimentos realizados para la construcción del detector de personas. En ese sentido, existen distintos métodos para fabricación de este tipo de detectores, por lo que es necesario escoger uno de ellos para su implementación en la RPI 3. Además, se deben detallar los factores tanto internos como externos del sistema que perjudican una correcta detección de las personas que transitan por un determinado lugar, de manera de lograr una alta tasa de aciertos en cuanto a las detecciones, y a su vez una baja cantidad de falsos positivos. De este modo se deberán realizar una serie de ajustes propios en el sistema para cumplir con las expectativas deseadas.

Un aspecto adicional que se ha decidido implementar en el detector, corresponde a contabilizar la cantidad de personas que logren ser detectadas por el sistema, de manera de realizar estudios estadísticos por parte de quien desee utilizar este tipo de aplicaciones, con el objetivo de presentar un valor agregado al trabajo realizado.

De manera preliminar se debe destacar que la construir del sistema propuesto debe ser de bajo costo y a su vez eficaz, utilizando los componentes disponibles en el laboratorio de robótica de la EIE-PUCV. Con estos antecedentes previos se han fijado una serie de objetivos para la realización del proyecto, los cuales se pasarán a revisar a continuación:

## Objetivo general

- Implementar un sistema que sea capaz de detectar personas en tiempo real, seguir su trayectoria y además de contabilizarlas, de manera de obtener algún tipo de estadística sobre las mismas. Para llevar a cabo la detección se utilizará el descriptor de histograma de gradientes orientadas, el cual debe ser implementado en un computador de placa reducida, siendo la principal alternativa la adaptación en una Raspberry Pi 3.

## Objetivos específicos

- Estudiar el estado del arte sobre la detección de personas mediante el descriptor HOG.
- Desarrollar de un sistema capaz de detectar personas utilizando alguna librería de procesamiento de imágenes, que permita realizar este tipo de tareas.
- Realizar pruebas de corta duración (no más de 10 minutos) con el detector, de manera que se adapte a las distintas condiciones a las que pueda ser sometido, sin perjudicar el desempeño del mismo.
- Realizar pruebas de larga duración (24 horas de grabación), para conocer la efectividad de las detecciones.
- Obtener una tasa de acierto de entre el 90% y el 95% en cuanto a las detecciones positivas de personas, y a su vez una baja cantidad de falsas detecciones.

# 1 Detección de personas

En este primer capítulo se introducirá al área relacionada con el procesamiento de imágenes y la visión por computador, debido a que se trabajará con imágenes digitales para realizar la detección de personas. Por este motivo es que además se presentarán algunos avances que se han desarrollado a lo largo del tiempo sobre estos temas, con el objetivo de interiorizar al lector a lo que será el posterior desarrollo del proyecto.

Es necesario conocer previamente el cómo se componen las imágenes, las cuales pueden ser definidas como funciones en dos dimensiones  $f(x_1, x_2)$ , donde  $x = (x_1, x_2)$  son las coordenadas espaciales y en donde el valor de  $f$  en cualquier  $x$  es la intensidad de la imagen en dicho punto. Desde este punto una imagen se puede definir como una función continua definida sobre un conjunto continuo (imagen analógica) o como una función discreta definida sobre un dominio discreto (imagen digital) [5].

## 1.1 Procesamiento digital de imágenes

El inicio del procesamiento de imágenes surge aproximadamente en la década de los años 20, pero su desarrollo sigue en constante crecimiento hasta el tiempos modernos [6], esto debido a que con la aparición del computador, se han logrado obtener mejores procesamiento de las imágenes, producto de la alta cantidad de recursos que esto requiere. Por lo tanto para poder entender el cómo se ha llegado a tales niveles de ejecución, es necesario conocer su origen.

El principal desarrollo en esta área comienza con el lanzamiento del primer satélite, el Landsat, por parte de la NASA en el año 1972, es ahí donde por parte de la comunidad científica dedicada al estudio de recursos naturales, implementando una nueva forma de manejar los datos obtenidos [7]. Por lo tanto, el proyecto satelital no solo trae beneficios para lo que será una nueva forma de manejo de estos datos, sino que también lo es con el trabajo sobre imágenes satelitales (como se muestra en la Figura 1-1), las cuales brindan una visión global del paisaje terrestre, las que nunca habían sido capturadas de tal forma. Cabe señalar que técnicas implementadas para el procesamiento de imágenes, fueron desarrolladas mucho antes de este evento, las cuales eran utilizadas en el área militar y científica [8].

Las imágenes se pueden clasificar de acuerdo a su fuente de obtención, esto puede ser en el rango visible del espectro electromagnético, el espectro IR (infra rojo), rayos X, etc. [9]. En ese sentido,

son muchos los procesamientos que se pueden realizar, pero para ello es fundamental obtener una imagen de entrada, luego se puede trabajar tanto en los colores de la misma, compresión, representación, reconocimiento, segmentación, etc.

El objetivo del procesamiento en las imágenes se basa principalmente en dos aspectos, el primero es mejorar la información contenida en una imagen para la interpretación humana, el segundo tiene relación con el tratamiento de los datos para la percepción autónoma para una máquina [2].

A mediados de la década de los 80 el costo de los equipos dedicados al procesamiento de imágenes era muy elevado, comparable con microscopios de alta definición o herramientas específicas para laboratorios, pero al comenzar a desarrollarse de manera más masiva los computadores, el estudio de las imágenes digitales comienza a propagarse significativamente. Además esto viene acompañado de mejoras en los procesadores fabricados, permitiendo un mejor y más rápido manejo de datos, ampliando los métodos empleados para el tratamiento de las imágenes [8].

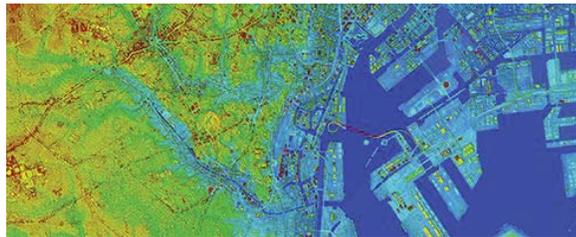


Figura 1-1: Procesamiento a una imagen satelital (fuente: <http://www.global-pasco.com>).

## 1.2 Visión por computador

La visión por computador está siendo explorada de forma exponencial debido a los avances tecnológicos que cada vez más van adquiriendo mayor importancia para el desarrollo de la inteligencia artificial e intenta emular la visión humana (Figura 1-2) utilizando diversas técnicas [10]. Tanto es así que posibles utilidades a las que se pueden aplicar estos desarrollos son variados, dependiendo del área de interés. Por ejemplo la detección de personas de a poco va adquiriendo importancia ya que se ha logrado implementar sistemas capaces de detectar, seguir y hasta reconocer rostros de una persona. Para ello es importante que la visión por computador busque solucionar problemas fundamentales, primero en mejorar la calidad de las imágenes para la interpretación humana y segundo, el procesamiento de datos de la escena el cual permita a las maquinas realizar una interpretación de forma autónoma [11].

En el campo de la visión por computador existen diversas aplicaciones, por ejemplo el seguimiento de objetos ya sea en video o tiempo real, el cual es la base de muchos sistemas como lo puede ser la robótica, juegos interactivos, navegación de vehículos, monitoreo de tráfico, además se puede emplear en otras áreas como por ejemplo para análisis de terreno, medicina, reconocimiento de rostros y huellas dactilares, entre muchas otras. Estos algoritmos utilizados deber ser rápidos y eficientes, y que no permitan grandes errores ni demoras [12]. En ese sentido,

su crecimiento continuo se debe, entre otras cosas, al desarrollo de distintos lenguajes de programación y sistemas operativos [6].

El método empleado para la detección de algún objeto en específico se basa en tres etapas claves: primero la obtención de una imagen para analizar, luego la localización del objeto a clasificar y por último la clasificación respectiva del(los) objeto(s), según sea el caso [13]. En ese sentido y para poder obtener una evaluación de forma concreta, es necesario realizar mediciones de forma manual de los objetos estudiados, en este caso corresponde a la detección de personas, de esta forma es posible determinar si el procesamiento efectuado está cumpliendo con los requerimientos estipulados.

Según la Asociación de Imágenes Automatizadas (AIA) [11], la visión por computador incluye todas las aplicaciones industriales y no industriales donde una combinación de hardware y software permiten una adecuada ejecución de sus funciones en base a la captura y el procesamiento de imágenes. A pesar de que la visión por computador de manera industrial utiliza muchos de los algoritmos y enfoques desarrollados en aplicaciones académicas-educativas, las limitaciones de esta última es evidente. Los sistemas implementados en la industria necesitan de mayor solidez, confiabilidad y estabilidad, en comparación a los sistemas desarrollados de manera académica, además su costo suele ser mucho menor [11].

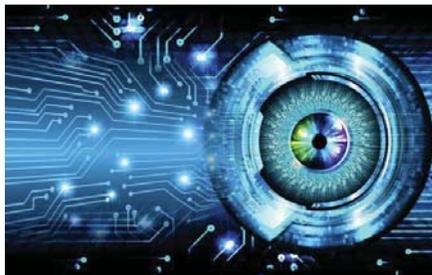


Figura 1-2: Visión por computador, intentando emular el ojo humano (fuente: <https://venturebeat.com>).

### 1.3 Detección de objetos

Existen variados métodos que permiten realizar detección de objetos (Figura 1-3), cada uno distinto del otro, los cuales siguen desarrollándose con el paso del tiempo. Por ejemplo se puede detectar la presencia de un objeto en movimiento a través de la sustracción de fondo [14], es una técnica utilizada para generar una máscara de primer plano, esto quiere decir que se obtiene una imagen binaria que contiene píxeles del objeto en movimiento, utilizando una cámara estática. También puede utilizarse el Filtro de Kalman [15], el cual tiene como propósito analizar una escena en movimiento para realizar una “predicción” sobre la posible siguiente ubicación del(los) objeto(s) que se desplaza(n), por lo tanto se espera que el movimiento del mismo sea continuo y por lo tanto predecible. En otras palabras lo que ocurre es que se buscan características como bordes, puntos, esquinas, región, etc., en una área determinada de la siguiente imagen alrededor de la sección predicha, teniendo la seguridad de que se lograrán encontrar características dentro de un cierto grado de confianza.

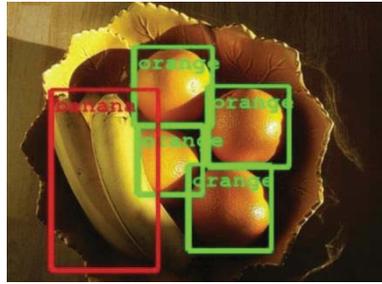


Figura 1-3: Detección y reconocimiento de objetos (fuente: <https://www.whatsnew.com>).

Otra manera de realizar detecciones corresponde al Histograma de Gradientes Orientadas (HOG, en su sigla en inglés) [16], el cual utiliza la técnica de ocurrencia de gradientes en porciones determinadas en una imagen, ventana de detección o región de interés, y luego estas gradientes calculadas son divididas en un conjunto de histogramas, los cuales permiten describir el objeto a analizar. Otra de las maneras de detectar objetos corresponde al de imagen en profundidad, para ello se requieren de sensores que permitan calcular la distancia entre estos sensores y un punto determinado en la escena, en ese sentido se emite un haz de luz a una determinada frecuencia, con el objetivo de diferenciar el objeto a analizar con el resto de la imagen [17].

### 1.3.1 Detección de personas

La capacidad de detectar a seres humanos mediante visión por computador es la clave para que computadores logren interactuar de manera inteligente y sin esfuerzos con un entorno rodeado de personas [2]. Si bien son muchas las aplicaciones que se han logrado realizar, el detectar y reconocer personas es una de los ámbitos más activos en la visión por computador [14].

Detectar seres humanos en una imagen es una tarea difícil debido a las apariencias que estos poseen, la cual es muy variable, además de la amplia gama de posturas que pueden adoptar, debido a esto es que se requiere implementar un sistema robusto, que permita identificarlas de manera clara e independiente, a pesar del escenario en el que se encuentre, ya sea por la diversidad de edificaciones, por la iluminación presente, o por otros factores externos que puedan afectar una detección.

Como se mencionó anteriormente, existen diversas técnicas para realizar una detección determinada sobre algún objeto, y dentro de esa gama de posibilidades, existen descriptores que permiten realizar el mismo trabajo para la detección de personas. Dentro de los descriptores más destacados se encuentra el LBP (Local Binary Patterns), el cual corresponde a un descriptor de texturas simples, invariante a cambios monótonos [18], otro descriptor es el SIFT (Scale-Invariant Feature Transform) [19], quien detecta y describe características locales en una imagen, otro método corresponde al SURF (Speeded-Up Robust Features) [20], quien está inspirado en SIFT, con la ventaja de ser más rápido y robusto que este último. Otro descriptor es el GLOH (Gradient Location and Orientation Histogram) [16], el cual es muy parecido al SIFT, con la diferencia que considera regiones más grandes para el análisis de los histogramas. Sin embargo distintos análisis y estudios realizados han determinado que la mejor técnica para la detección de personas

corresponde al descriptor HOG (Histogram of Oriented Gradients) [16], que como se mencionó anteriormente utiliza la técnica de ocurrencia de la gradiente, en sectores determinados de una imagen.

## 1.4 Problemática

Como se ha mencionado en los apartados anteriores, la visión por computador está en constante desarrollo, ya que el avance tecnológico en cuanto a dispositivos electrónicos, ha permitido ir mejorando las técnicas utilizadas en el procesamiento digital de imágenes para lograr la semejanza con el ojo humano. En ese sentido, la detección de personas se está volviendo cada vez más importante para temas relacionados con seguridad o para simplemente obtener datos estadísticos sobre el flujo de las mismas en situaciones determinadas, ya que la visión por computador, en algunos casos, puede lograr mejores resultados que los trabajos realizados por la inspección de una persona, ya que un largo proceso de análisis, implica un cansancio en la persona, afectado en algunos casos a la visión del mismo, afectando los resultados obtenidos para las pruebas.

Por este motivo es que con el paso del tiempo ha tomado mayor importancia el desarrollar un sistema capaz de detectar personas, que sea de bajo costo y a la vez efectivo. Razón por la cual, los computadores de placa reducida van adquiriendo mayor importancia, ya que su costo es mucho menor al de un computador convencional, además, el tamaño de la placa es mucho menor en comparación a la de un computador, permitiendo adaptarlo a distintas situaciones para la realización de tareas. A su vez, existen aplicaciones que no requieren de una gran cantidad de recursos, por lo que las características de un computador de placa reducida permitirían el desarrollo de proyectos, presentando los mismos resultados satisfactorios que los de un computador convencional.

Es necesario señalar que la detección de personas es muy conocido dentro del área el procesamiento de imágenes, la cual es realizada mediante algún tipo de descriptor , en donde existen, además, distintos lenguajes de programación aplicados a diversos objetivos específicos, por lo que no es un tema nuevo a desarrollar. Sin embargo, la mayoría de estas aplicaciones son implementadas en computadores de escritorio, solo unas pocas son aplicables a computadores de placa reducida, principalmente por no contar con las características técnicas necesarias, para un buen desempeño del programa.

Si bien existe un gran número de computadores de placas reducidas, por ejemplo RPi, Arduino, Edison, etc. No todas poseen las mismas características, por lo que se hace necesario conocer las distintas alternativas presentes en el mercado, de manera de elegir la que mejor cumpla con los requerimientos deseados, y en caso de ser necesario, adaptarse a las características que posea esta, para obtener un adecuado funcionamiento de la placa, al momento de realizar distintos procesamientos digitales en las imágenes.

## 1.5 Solución

El método empleado para el desarrollo del proyecto, en cuanto a la detección de personas, corresponde al descriptor HOG, ya que es uno de los métodos más efectivos para este tipo de implementaciones, debido a que solo se pretende obtener información sobre las siluetas detectadas, de manera de clasificarlas como persona o no.

Cabe destacar que a pesar de tratarse de uno de los mejores métodos para la detección de personas, HOG no está libre de errores, ya que son diversos los factores que pueden afectar la detección correcta de una persona, por lo que puede requerir ciertos ajustes para obtener un óptimo desempeño.

Siguiendo el propósito del proyecto, es importante conocer las distintas aplicaciones que se han desarrollado con este tipo de algoritmo en computadores de placa reducida, ya que a pesar de pretender elaborar un sistema de fácil instalación y utilización, este debe contar con una confiabilidad importante. Motivo por el cual, es que se ha optado por utilizar una Raspberry Pi, siendo una de las placas con mejores características en cuanto a la relación precio-calidad que se puede encontrar en el mercado, además ya se tiene algunos registros de aplicaciones que se han podido realizar en ella [21].

### 1.5.1 Sistema desarrollado

Debido a esta problemática presentada y a la solución propuesta, es que se ha decidido implementar un detector de personas en una RPI 3, a la cual se le ha decidido instalar la librería OpenCV para realizar el procesamiento en las imágenes, ya que es una librería de código abierto, de fácil acceso y configuración.

La detección de personas tiene por objetivo determinar la presencia o no de ellas, al momento de capturar una imagen, por lo tanto, en caso de ser efectivo, se debe indicar su posición en la escena (Figura 1-4). En ese sentido, para entender el funcionamiento del detector se deben analizar dos etapas; la primera corresponde a la obtención de una imagen a estudiar, y la segunda tiene relación con la clasificación de la misma, precisamente para la extracción de información es que se utiliza HOG.

### 1.5.2 Computadores en placas reducidas

Un computador de placa reducida, es un computador de placa única [22], esto quiere decir que todos los componentes de un computador tradicional, se encuentran integrados en una sola placa. Un computador de escritorio o un computador portátil, generalmente tiene componentes separados, que están conectados por cables a una placa de circuito central [23].

Un computador de una sola placa, contiene el microprocesador, el almacenamiento y la memoria, así como otros componentes necesarios, en una placa de circuito única, reduciendo considerablemente el tamaño que físicamente utiliza un computador convencional.

Existe una gran variedad de estos tipos de computadores, dependiendo de las necesidades que se requieren, aunque poseen una velocidad de ejecución de tareas más lento y limitado, en comparación a un computador personal, esto debido a que se utilizan para tareas simples y específicas. Es por este motivo que su uso se encuentra en aplicaciones de control de proceso, sistemas robóticos, entre otros [20].

La clasificación de personas ya es posible realizarla en computadores de placas reducidas [22], logrando resultados positivos, en cambio la detección de personas solo se ha implementado en computadores de escritorio, ya que se requiere de una gran velocidad de procesamiento, por lo que es necesario experimentar en este tipo de placas, si es factible en primer lugar utilizar las librerías deseadas para el procesamiento de imágenes, y en segundo lugar lograr realizar este tipo de aplicaciones en ella.

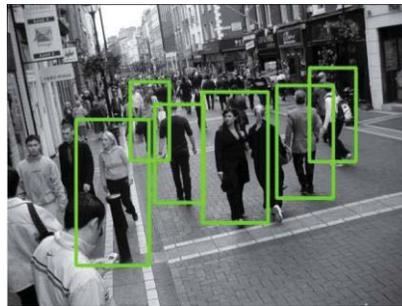


Figura 1-4: Detección de personas mediante descriptor HOG (fuente: <https://www.nextbigfuture.com>).

### 1.5.3 Librería OpenCV

OpenCV es una librería de visión por computador de código abierto, la cual está escrita en C y C++, además es compatible con sistemas operativos como Linux, Windows e IOS, a su vez cuenta con un desarrollo activo en interfaces Python, Ruby, Matlab, entre otros lenguajes. Por otro lado se encuentra orientado hacia aplicaciones en tiempo real [24].

Uno de los objetivos por los que fue desarrollado OpenCV es proveer una infraestructura de visión por computador, que sea fácil de utilizar y que permitiera acelerar el uso de la percepción de las maquinas en productos comerciales. Por otro lado busca realizar un trabajo de forma eficiente en cuanto al gasto de recursos computacionales, además intenta aprovechar los múltiples núcleos disponibles en los procesadores [25].

Esta librería contiene alrededor de 500 funciones que involucran distintas áreas de la visión por computador, ya que es posible trabajar junto con aprendizaje automático. Es más, se cuenta con una librería de uso general sobre este campo. Estas funciones permiten, entre otras cosas, reconocer rostros de personas, identificar objetos, rastrear movimiento, entre muchas otras aplicaciones [24].

## 2 Descriptor HOG

Uno de los principales métodos empleados en la visión por computador para la detección de personas corresponde al descriptor HOG, para lo cual se definirá lo que es un descriptor, un histograma, en qué consisten una gradiente y su respectivo cálculo, además de cómo es posible obtener una gradiente orientada, de esta forma conocer de manera general el funcionamiento del respectivo descriptor.

Los resultados de esta investigación se dieron a conocer el año 2005 en Francia por Navneet Dalal y Bill Triggs [16], para ello se basaron en distintos algoritmos, los cuales fueron mencionados anteriormente y que permiten realizar detección de objetos, pero más específicamente en este caso, la detección de personas. Esta nueva forma de detección debía ser simple y rápida, por lo tanto para la clasificación de la presencia o no de personas se utilizó la Máquina de Vectores de Soporte (SVM). Para el estudio se seleccionaron 1239 imágenes con presencia de personas en distintas posiciones, las cuales se duplicaron y reflejaron aumentando así este número al doble. Por contraparte se contó con 1218 imágenes libre de personas, estas fueron utilizadas para la creación de una base de datos. De esta forma, luego de realizar todo el proceso de cálculo de las gradientes, se procede a clasificar las imágenes mediante la SVM [16].

### 2.1 Descriptor de características

Un descriptor de características es una representación de una imagen la cual es simplificada mediante la extracción de información útil y se deshace de la información innecesaria. Esto permite generalizar la forma de un objeto deseado, con el objetivo de poder identificarlo al momento de extraer sus características, a pesar de encontrarse en distintas condiciones, por ejemplo en el caso de personas se identifica la cabeza, brazos, piernas, etc.

Existen descriptores locales como también globales [26][27], los cuales son utilizados dependiendo de la necesidad para lo que se desea estudiar, esto debido a que los descriptores locales describen puntos claves en la imagen, por lo que son utilizados para aplicaciones de nivel superior. Por contraparte, los descriptores globales describen una imagen como un todo, de manera de generalizar al objeto estudiado, esto permite obtener representaciones de los contornos o características de texturas, siendo ideales para aplicaciones de bajo nivel [26][27].

Por lo tanto, los descriptores locales son utilizados idealmente para el reconocimiento de objetos, y para la detección de clasificación de estos se utilizan los descriptores globales, HOG es un claro ejemplo de este tipo de descriptores. Sin embargo, es posible combinar estas dos características para mejorar la precisión en el reconocimiento de objetos, por lo que sería necesario contar con un computador que tenga un mínimo de características para lograr un buen rendimiento en el programa [26][27].

Para el descriptor de características en HOG, la información que es “útil”, tiene relación con la distribución de las direcciones de las gradientes. En ese sentido, las gradientes en una imagen permiten diferenciar bordes y esquinas, ya que estos puntos contienen mucha más información que las regiones planas, producto de los cambios abruptos en la intensidad de sus píxeles [26].

## 2.2 Histograma en una imagen

El histograma corresponde a un gráfico que permite conocer la distribución de frecuencias sobre un conjunto de datos [28]. Está compuesto por dos ejes, el eje X, que contiene un evento cuya frecuencia se desea contabilizar, y el eje Y, que contiene la frecuencia.

El histograma en una imagen (Figura 2-1), corresponde a un gráfico que muestra la distribución de intensidades de los píxeles, en una imagen en escala de grises [28][29]. Por lo tanto, en un histograma de una imagen, se encuentra en el eje X la intensidad de los píxeles, en algunos casos se presentan divididos en intervalos, llamados *bins* o contenedores, los que son determinados por el programador según lo estime conveniente, y en el eje Y se encuentra la frecuencia de las intensidades, la cual es representada por columnas verticales o barras, entre los que no se presenta separación, en comparación a los gráficos de barra, puesto que se representa a un conjunto de datos continuos [29].

Los valores en la intensidad van desde el color negro, que se encuentra al extremo izquierdo del gráfico, hasta el color blanco, al extremo derecho del mismo. Por lo tanto, una imagen al contener 8 bis por píxel [28], significa que tiene 256 tonos de grises, partiendo desde el 0 (color negro), hasta el 255 (color blanco). Este rango se divide en intervalos (contenedores) equivalentes [29], los que son definidos por el usuario.

Las imágenes a color utilizan los mismos conceptos de muestreo, cuantificación y resolución que las imágenes en escala de grises. Sin embargo, en lugar de un único valor de intensidad que se expresa en el nivel de grises, los píxeles de las imágenes a color están cuantificados utilizando tres componentes, los que poseen distinto significado según el espacio de color utilizado. Un espacio de color es una representación matemática del conjunto de colores que permite la especificación de un color de manera estándar, el más conocido es el RGB. En este modelo, cada color se encuentra descompuesto en sus tres componentes espectrales primarios Rojo, Verde y Azul, por lo tanto su histograma se encuentra representado por tres gráficas [29].



Figura 2-1: Representación de un histograma en una imagen (fuente: <https://djiphotoacademy.com>).

## 2.3 Detección de bordes

La detección de bordes es una técnica para conocer los límites de los objetos dentro de una imagen [30], por lo tanto, permite encontrar discontinuidades en el brillo. Este método permite realizar segmentación de la imagen y extracción de datos para el posterior procesamiento de la imagen.

Los algoritmos más comunes para la detección de bordes son: Sobel, Canny, Roberts y métodos de lógica difusa [30].

Los bordes y las líneas son cambios repentinos en los niveles de escala de grises, en una pequeña distancia espacial. Por otra parte, es necesario distinguir los bordes, con pequeñas características de la imagen, como lo son las texturas o el ruido de la misma, de manera de detectar todos los bordes, mientras se elimina el ruido presente [31].

La mayoría de las técnicas para detectar bordes emplean operadores locales basados en la primera y segunda derivada de los niveles de grises de una imagen. La derivada de una señal continua indica las variaciones locales con respecto a una variable, de forma que el valor de la derivada es mayor cuanto más rápido son las variaciones. En los casos bidimensionales  $f(x,y)$ , la derivada es un vector que apunta en la dirección de la máxima variación de  $f(x,y)$ . En el caso de bidimensional discreto, las distintas aproximaciones del operador gradiente de basan en las diferencias entre los niveles de grises de la imagen.

## 2.4 Gradiente

La gradiente es un vector que indica magnitud y dirección, por lo tanto en una imagen se presenta como variación en la intensidad de los píxeles adyacentes tanto en la dirección horizontal, como en la dirección vertical. Para realizar el cálculo de la gradiente se debe tener en consideración lo siguiente:

Se debe aplicar un filtro a la imagen utilizando dos máscaras, una para calcular la componente horizontal  $G_x$  y otra para la componente vertical  $G_y$  de la gradiente. Las máscaras utilizadas son  $maskH = [-1 \ 0 \ 1]$  y  $maskV = [-1 \ 0 \ 1]^T$  respectivamente. Para el cálculo de la magnitud se utiliza la ecuación (2-1) y para el ángulo la ecuación (2-2):

$$|G(m,n)| = \sqrt{G_x^2(m,n) + G_y^2(m,n)} \quad (2-1)$$

$$\varphi G(m,n) = \tan^{-1}(G_y(m,n)/G_x(m,n)) \quad (2-2)$$

A continuación, en la Figura 2-2, se presentará un ejemplo para un mejor entendimiento del cálculo de la gradiente, en donde se ha elegido un pixel al azar, para obtener su gradiente orientada. La Figura 2-2a) corresponde a la imagen original a la que se realizará el cálculo de una gradiente orientada, la Figura 2-2b), a su vez, corresponde a una región de 8x8 píxeles que será analizada, por último la Figura 2-2c) corresponde a los valores de cada píxel de b), de la cual se realizaran los cálculos según corresponda.

Primero se restan los valores de los píxeles en la dirección del eje X, en este caso se restarán de derecha a izquierda, por lo tanto quedará:  $(168 - 142 = 26)$ , esto mismo se realiza en la dirección Y, esta vez desde arriba hacia abajo:  $(178 - 149 = 29)$ . Con estos dos valores se puede obtener el vector de gradiente:  $[26 \ 29]^T$ .

Cabe mencionar que el orden puede ser de derecha a izquierda y de arriba hacia abajo o viceversa, solamente hay que mantener el mismo orden para cada uno de los píxeles analizados. Además, que su valor sea negativo no tiene mayor injerencia, ya que para la magnitud los valores están elevados al cuadrado, por lo que siempre serán positivos. En el caso de la fase, los ángulos varían de  $0^\circ$  a  $180^\circ$ , en vez de  $0^\circ$  a  $360^\circ$ . Esto ya que se utilizan las gradientes “sin signo”, esto quiere decir que una fase negativa se considera igual a su opuesto a los  $180^\circ$  [32].

Este proceso se realiza para cada uno de los píxeles que componen la celda, por lo tanto se realizan 64 cálculos de gradientes, lo que posteriormente ocurre, es posicionar estos 64 datos en los 9 contenedores que conforman el histograma, esto permite reducir de 64 a tan solo 9 “flechas” que componen a la celda. Este mismo proceso debe realizarse para cada una de las celdas que componen la imagen.

Magnitud:  $\sqrt{26^2 + 29^2} = 38,98$

Fase:  $\tan^{-1}\left(\frac{29}{26}\right) = 48,12^\circ$

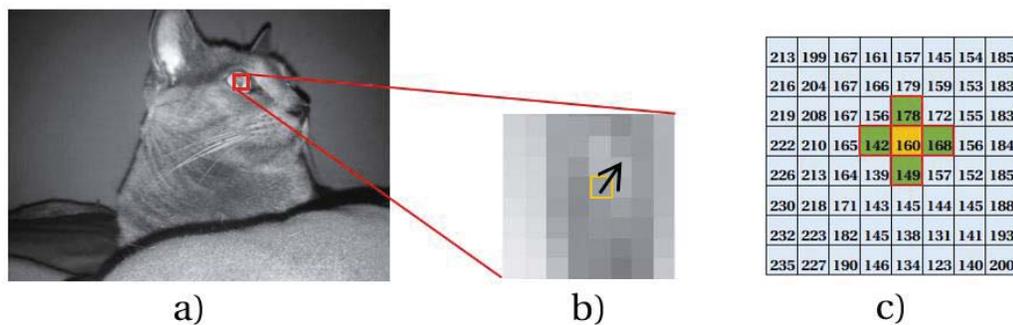


Figura 2-2: a) Imagen original analizada, b) Región analizada y el resultado del cálculo de la gradiente, c) Valores de cada uno de los píxeles, en donde la región amarilla corresponde al píxel analizado y las regiones verdes a las respectivas máscaras.

## 2.5 Descriptor HOG

Es un descriptor utilizado para la detección de objetos en visión por computador y para el procesamiento de imágenes. La técnica consiste en ocurrencias de gradientes en porciones localizadas de una imagen, ventana de detección o región de interés. En otras palabras, la imagen es dividida en un conjunto de celdas uniformes (Figura 2-3). El algoritmo estima la orientación de gradientes de los píxeles que forman la celda y reúne la información en un histograma de orientaciones de  $N$  clases. Este conjunto de histogramas se utiliza para describir el objeto a utilizar.

El algoritmo que contiene los descriptores HOG, tiene como entrada una imagen a color seleccionada previamente por el usuario. De esta forma se busca estandarizar el tamaño de entrada de la imagen, por una nueva de  $64 \times 128$  píxeles, esta selección se conoce como Región de Interés (ROI). Este nuevo tamaño se utilizará a lo largo de todo el algoritmo para obtener los descriptores. Luego de esto se procede a convertir la imagen a color, en una imagen a escala de grises, sobre esta imagen se estiman los componentes horizontal y vertical del gradiente utilizando las máscaras *maskH* y *maskV* respectivamente [16].

El proceso consiste en obtener los componentes del gradiente sobre cada uno de los píxeles de la imagen para luego calcular la magnitud y el ángulo. A su vez, cada magnitud es almacenada en 1 de 9 diferentes matrices de acuerdo a su correspondiente ángulo. Por otra parte el intervalo de ángulos se encuentra entre los  $0^\circ$  a  $180^\circ$ .

Luego de obtener cada una de las orientaciones para cada píxel, se procede a dividir la imagen en celdas uniformes no traslapadas. Las posibles cantidades de celdas son potencias de 4, vale decir 1, 4, 16, 64, hasta 4069. Hay que tener en consideración que el número de filas y columnas por celdas deben ser iguales.

Cada una de las celdas entrega 9 descriptores, los que corresponden a la sumatoria de las magnitudes en cada uno de los segmentos. En una imagen se puede calcular el total de descriptores (TD) que la componen, en donde  $TD = NC \times 9$ ,  $NC$  a su vez corresponde al número de celdas utilizadas, las que son definidas previamente por el usuario. En otras palabras, los descriptores HOG de una imagen son entregados en una vector de TD elementos, donde cada 9 elementos del vector, se representa una celda.

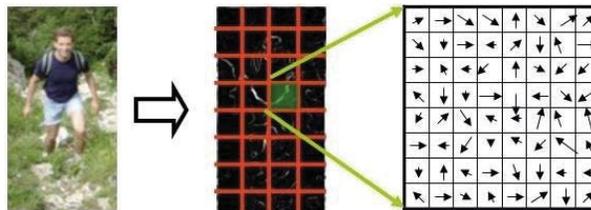


Figura 2-3: Orientación de gradientes en una celda (fuente: <https://www.quora.com>).

Posterior a determinar los descriptores HOG que representan a la persona, es necesario construir una relación entre la persona y su posible ubicación en un cuadro siguiente, esto a partir de una medida de disimilitud entre conjunto de descriptores.

La idea principal de un descriptor HOG es que la apariencia y la forma determinada de un objeto en una imagen, pueda ser descrita por la distribución de la intensidad de gradientes o direcciones de contornos. Por lo tanto, la combinación de los histogramas forma lo que es el descriptor. Para obtener mejores resultados, los histogramas locales están normalizados mediante el cálculo de una medida de intensidad (Energía) en áreas más extensas que las celdas (Bloques), de esta forma es posible normalizar todas las celdas en un bloque. Por medio de esta estandarización se logra una mejor resistencia a los cambios de luminosidad. En los experimentos realizados por Dalal y Triggs [16] se utilizaron 2 tipos de geometría para la detección, la rectangular (R-HOG) y la circular (C-HOG), como se muestra en la Figura 2-4. Para el análisis con la configuración rectangular se utilizaron bloques de tamaño de 3x3 celdas y celdas de 6x6. Pero estos son valores referenciales ya que sin problemas se pueden utilizar bloques de 2x2 celdas y celdas de 8x8 píxeles, obteniéndose los mismos buenos resultados que con la configuración anterior, todo depende del usuario que esté programando.

Existe una configuración del bloque de forma circular, que se conoce como C-HOG, la cual se comparará con la configuración rectangular R-HOG. R-HOG es muy similar al funcionamiento del descriptor SIFT, pero funcionan de manera distintas, el primero analiza todo el conjunto de la imagen, mientras que el segundo se centra en puntos en específico [16].

Para la detección humana, se utilizó  $3 \times 3$  bloques de celdas. De las celdas de  $6 \times 6$  píxeles tienen el mejor rendimiento, con una tasa de error del 10,4%, las celdas de 6 a 8 píxeles de ancho son las mejores independientemente del tamaño del bloque. También se varió el tipo de tamaño del bloque, ya sea un bloque vertical (2x1 celdas) y bloque horizontal (1x2 celdas) y una mezcla de ambos. Con la mezcla de ambos bloques de obtuvieron mejores resultados que cada uno por sí solo, pero no fueron tan efectivos como lo son los bloques de 2x2 celdas [16].

C-HOG es una configuración que posee dos opciones, una presenta solo una celda central y la otra con una celda central angularmente dividida, los resultados obtenidos proporcionan el mismo rendimiento.

En el trabajo realizado por Dalal y Triggs [16], no se detalla cuál posee mejor rendimiento, R-HOG o C-HOG, pero independiente de la imagen a analizar, la mayoría de los descriptores estudiados han adoptado la configuración R-HOG, por lo que se trabajará con el mismo método.

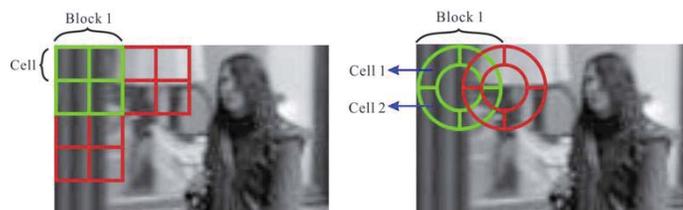


Figura 2-4: R-HOG vs C-HOG (Fuente: <http://hli2020.github.io>).

### 2.5.1 Construcción de HOG

Para realizar el cálculo de las gradientes orientadas, lo que primero se debe hacer es seleccionar una región de interés sobre la imagen a analizar, con un tamaño recomendado de 64x128 píxeles, esta “nueva imagen” se divide en celdas compuestas de 8x8 píxeles a las cuales se les calcula su gradiente, luego se fabrican bloques de 2x2 celdas, vale decir 16x16 píxeles (de esta forma se puede calcular que habrá un total de 105 bloques (7x15) en la imagen). Posteriormente se normaliza el histograma obtenido para cada uno de las celdas (Figura 2-5).

En el caso de los bloques, estos cuentan con un solapamiento del 50% entre ellos, de esta forma se permite que el descriptor posea una mayor tolerancia a los cambios de iluminación y sombreado de la imagen analizada [13]. Al analizar los píxeles que rodean directamente al deseado y repetir este paso para cada pixel de la imagen resulta muy beneficioso, ya que si solo se comparan los píxeles sin realizar una comparación, imágenes muy claras o muy oscuras de la misma persona, tendrán distintos valores para en los píxeles. Por lo que al considerar la dirección del brillo, la representación es independiente de la tonalidad de la misma, por este motivo es que se representa con una flecha la dirección de la gradiente. Guardar la información de cada píxel, es una manera muy detallada de analizar la imagen, es por este motivo que resulta más beneficioso segmentar la imagen en cuadros de 16x16 píxeles cada una, y en cada cuadro se cuenta el número de gradientes que se encuentran allí [17].

Para aumentar la precisión en la detección, es aconsejable normalizar la imagen de entrada antes de la descripción, para ello, y según lo comprobaron Dalal y Triggs [16], la normalización mediante el método de la raíz cuadrada ayuda en obtener mejores resultados. Este método consiste en calcular la raíz cuadrada de cada píxel de la imagen, de esta forma se comprime la intensidad de los píxeles.

Otra aplicación del descriptor HOG es para el análisis de objetos, como por ejemplo el reconocimiento de logos (Figura 2-6), ya que es posible cuantificar y representar formas y texturas de una imagen. Para llevar a cabo la detección, es necesario utilizar el descriptor junto con un clasificador, en este caso SVM, más una base de datos con imágenes del objeto a analizar [33].

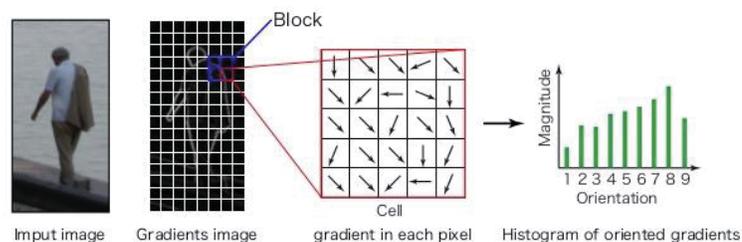


Figura 2-5: Construcción del descriptor HOG (fuente: <https://www.researchgate.net>).

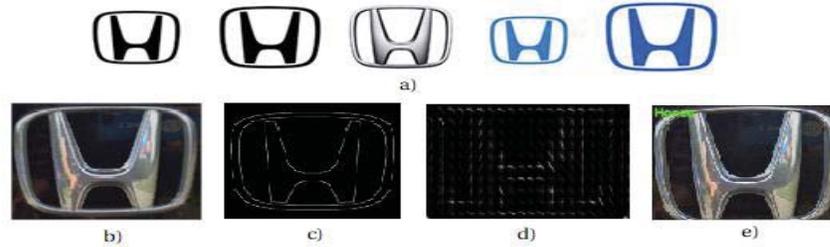


Figura 2-6: a) Imágenes de entrenamiento, b) Imagen a analizar, c) Obtención de su contorno, d) Representación en HOG, e) Resultado obtenido para la clasificación (fuente: <https://gurus.pyimagesearch.com>).

### 2.5.2 Esquema de la construcción del descriptor

En primer lugar, se debe seleccionar una imagen que se desea analizar para realizar una detección, en este caso corresponde la detección de una persona, luego es necesario normalizar esta imagen, esto quiere decir que se realiza un “ajuste” en el histograma, permitiendo distinguir con mejor claridad los cambios de intensidad en los píxeles correspondiente a los distintos bordes presentes, de esta forma se procede a calcular las gradientes orientadas en toda la imagen. Posterior a ello, se realiza la separación en celdas de la imagen de entrada, las que tiene una dimensión de 8x8 píxeles, para luego conformar los bloques de 2x2 celdas, las que serán superpuestas al momento de realizar un deslizamiento en la ventana, permitiendo normalizar los histogramas obtenidos. Es así como los datos obtenidos son “recolectados” para un posterior análisis en una máquina de vector de soportes y finalmente poder clasificar la imagen analizada, con la presencia o no de alguna(s) persona(s), en caso de existir la presencia de una persona, esta debe ser señalada mediante un rectángulo de color verde.

A continuación se presentará un esquema (Figura 2-7) con la construcción del descriptor HOG, y los resultados que se obtienen en cada paso descrito (Figura 2-8), para un mejor entendimiento del mismo.

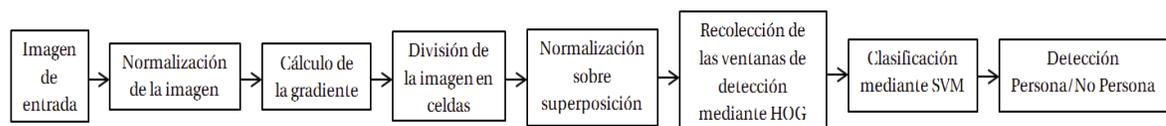


Figura 2-7: Diagrama de bloques para la construcción de HOG.

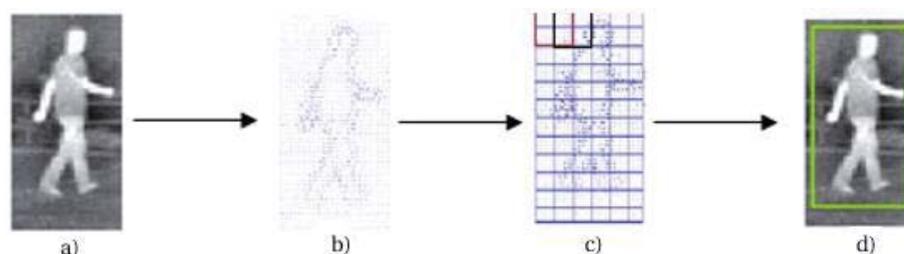


Figura 2-8: a) imagen de entrada, b) cálculo de gradiente, c) división de celdas d) detección de persona (fuente: <https://www.researchgate.net>).

## 2.6 Máquina de vector de soportes

La máquina de vectores de soporte surge en los años 90 basados en la teoría del aprendizaje estadístico. En un principio las SVMs fueron pensadas para resolver problemas de clasificación binaria, hoy en día se utilizan para diversos tipos de problemas, por ejemplo agrupamiento o multclasificación y se ha utilizado también, con éxito, en el campo de la visión por computador.

Dentro de los clasificadores, los SVMs pertenecen a la categoría de clasificadores lineales, ya que introducen separadores lineales o hiperplanos, ya sea en el espacio originales de los ejemplos de entrada, en el caso de que sean separables o cuasi-separables, o en un espacio transformado, si los ejemplos no son separables de forma lineal en el espacio original, donde de ser así se utiliza una función llamada kernel [34].

La idea de este método es reducir el error en la clasificación, para ello se selecciona un hiperplano de separación que sea equidistante a los ejemplos más cercanos de cada clase, de forma de conseguir un margen máximo a cada lado del hiperplano. Es necesario señalar que al momento de generar el hiperplano, son considerados los ejemplos de entrenamiento de cada clase que se encuentran en la frontera de dichos márgenes. Estos ejemplos reciben el nombre de vectores de soporte. Esta es la principal diferencia con la mayoría de los distintos clasificadores existentes, los cuales buscan minimizar los errores cometidos por el modelo generado a partir de los ejemplos de entrenamiento [34].

En esta sección se pretende dar a conocer conceptos generales en la construcción de una SVM, por lo que no se ahondará en mayores detalles ya que no es un objetivo de este trabajo, sin embargo es un concepto que se debe conocer ya que se utiliza en la clasificación de personas del descriptor HOG.

A continuación se detallarán los posibles métodos de clasificación para clases binarias, esto quiere decir, primero lograr separa correctamente las clases de datos en el entrenamiento, presentes en un plano  $(x,y)$ , para posteriormente definir si pertenecen o no al conjunto determinado.

### 2.6.1 SVM para clasificación binaria para ejemplos linealmente separables

Dado un conjunto de ejemplos separables  $S = \{(x_1, y_1), \dots, (x_n, y_n)\}$ , donde  $x_i \in \mathbb{R}^d$  e  $y_i \in \{+1, -1\}$ , se puede definir un plano de separación como una función lineal, ecuación(2-3), capaz de separar dicho conjunto sin error.

$$D(x) = (w_1x_1 + \dots + w_dx_d) + b = \langle w, x \rangle + b \quad (2-3)$$

Donde  $w$  y  $b$  son coeficientes reales. El hiperplano de separación debe cumplir con las siguientes restricciones, ecuación (2-4) para todo  $x_i$  del conjunto de ejemplos:

$$\begin{aligned} \langle w, x_i \rangle + b &\geq 0, & \text{si } & y_i = +1 \\ \langle w, x_i \rangle + b &\leq 0, & \text{si } & y_i = -1, i=1, \dots, n \end{aligned} \quad (2-4)$$

O de forma más compacta de escribir la ecuación anterior, es como la ecuación 2-5:

$$y_i (w \cdot x_i) \geq 0, \quad i=1, \dots, n \quad (2-5)$$

Como se puede apreciar en la Figura 2-9, el hiperplano que se genera para separar las clases no es solo uno, sino que existen infinitas posibilidades que permiten una correcta separación y que cumplen con las restricciones 2-3 y 2-4, es por ello que se ha establecido un criterio que permita definir el hiperplano óptimo [34].

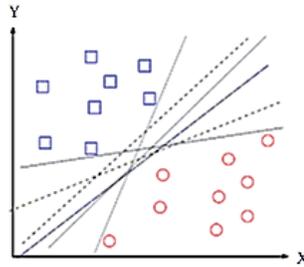


Figura 2-9: Infinitas posibilidades de hiperplano de separación [34].

### 2.6.2 SVM para clasificación binaria de ejemplos cuasi-separables linealmente

Lo planteado en la sección anterior, tiene poco interés práctico ya que los problemas a resolver no son de forma ideal, sino que presentan dificultad para separar las clases de manera lineal, como se presenta en la Figura 2-10. La forma de resolver este tipo de problemas más parecidos a la realidad, es presentando un mayor grado de tolerancia a la separación del conjunto de ejemplos, permitiendo así, que se encuentren errores de clasificación en algunos ejemplos del conjunto de entrenamiento. No obstante, el objetivo sigue siendo el de encontrar un hiperplano óptimo para el resto de los ejemplos que sí son separables [19]. Para que un ejemplo sea linealmente separable debe cumplir con las siguientes características de la ecuación 2-6:

$$y_i (w \cdot x_i + b) \geq 1, \quad i=1, \dots, n \quad (2-6)$$

De no cumplir con dicha restricción, se habla de un sistema no-separable, por lo tanto se pueden presentar dos casos. El primero corresponde a que el ejemplo cae dentro de los márgenes asociados a la clase correcta, según la frontera de decisión definida por el hiperplano de separación. En el segundo caso, el ejemplo se encuentra al otro lado de dicho hiperplano. En ambos casos se dice que el ejemplo es no-separable, independiente que en el primer caso la clasificación sea correcta, ya que la otra clasificación resultó errónea [34].

Para solucionar el problema presentado, se requiere introducir a la ecuación 2-6 un conjunto de variables reales positivas denominadas *variables de holgura*,  $\xi_i, i=1, \dots, n$ , que permiten cuantificar el número de ejemplos no-separables que se está dispuesto a admitir, esto es posible visualizarlo en la ecuación 2-7:

$$y_i (\langle w, x_i \rangle + b) \geq 1 - \xi_i, \quad \xi_i \geq 0, i=1, \dots, n \quad (2-7)$$

Al igual que en el caso anterior no se entrará en más detalles sobre el tema, solo se mencionará que el hiperplano formado ya no es de forma lineal, sino que es curvo, siguiendo la ideal de una separación lo más exacta posible [34].

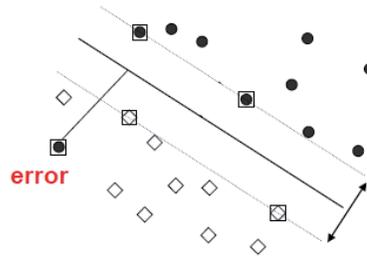


Figura 2-10: Ejemplo de un hiperplano no lineal para la clasificación de conjuntos no-separables [34].

### 2.6.3 SVM para clasificación binaria de ejemplo no separables linealmente

Las dos opciones presentadas anteriormente se analizan para hiperplanos en donde la clasificación de los ejemplos es perfectamente separable o cuasi-perfectamente separable. Por este motivo es que los hiperplanos pueden definirse como funciones lineales de los ejemplos. En este caso es necesario definir un conjunto de funciones base, no lineales (Figura 2-11), para definir el espacio transformado de alta dimensionalidad, además de cómo buscar un hiperplano de separación óptimo de dichos espacios transformados [34].

La función de decisión viene dada por la expresión 2-8:

$$D(x) = \sum a_i * y_i K(x, x_i) \quad (2-8)$$

Donde  $K(x, x_i)$  se denomina Función Kernel.

Por definición la función Kernel es una función  $K: \mathbb{X} \times \mathbb{X} \rightarrow \mathbb{R}$  que asigna a cada par de elementos del espacio de entrada  $\mathbb{X}$ , un valor correspondiente al producto escalar en las imágenes de dichos elementos.

Existen diferentes tipos de funciones Kernel, dependiendo de la necesidad del usuario para la clasificación, por ejemplo se encuentra la función kernel lineal, polinómico de grado  $p$ , la función de base radial, kernel gaussiano, la red neuronal sigmoideal, la inversión multicuadrática. Cada una de ellas presenta un mejor rendimiento dependiendo de la distribución de las clases que se desean clasificar, por lo que es necesario conocer para qué función es más óptima la utilización de cierta función kernel [34].

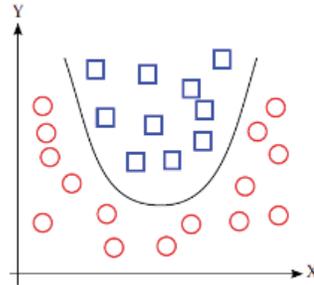


Figura 2-11: Clasificación binaria no separable linealmente [34].

La elección de una de estas configuraciones es muy relevante al momento de desear un buen clasificador, es por este motivo que para la detección de personas en el trabajo realizado por Dalal y Triggs [16], se utilizó la última de las configuraciones expuestas anteriormente, ya que son muchas las siluetas que pueden llevar al error al momento de querer clasificar una escena, permitiendo así lograr crear un sistema robusto el cual, a pesar de contener una gran cantidad de imágenes para su entrenamiento, está expuesto a presentar errores en cuanto a la clasificación ya sea de una silueta u objeto que pueda ser asimilado a una persona.

OpenCV cuenta con una función que además de contar con la base de datos [16] utilizadas para el entrenamiento del sistema para la detección de personas, trae implementada la clasificación por SVM, tal como lo realizado por Dalal y Triggs [16], no obstante, la exactitud en la clasificación puede ser ajustada por el programador según lo estime conveniente, ya que depende de la implementación que se desee darle al detector.

Por otra parte si el programador no se encuentra conforme con los parámetros utilizados, en cuanto al tamaño (resolución) de las imágenes de entrenamiento, por ejemplo, para realizar la clasificación de las personas, perfectamente puede crear su propia base de datos, las cuales contengan todas las modificaciones que desee realizar en las imágenes, permitiendo obtener un sistema que logrará resultados positivos para una determinada situación a la que se encuentre expuesta.

Debido a lo planteado anteriormente, es necesario conocer aunque sea de forma muy superficial, el cómo se logran obtener clasificaciones mediante SVM, permitiendo realizar modificaciones coherentes y que sean beneficiosas para, en este caso, la clasificación de personas.

## 3 Raspberry Pi

La utilización de computadores de placa reducida ha aumentado significativamente con el paso del tiempo ya que sus características son muy similares a las de un computador de escritorio, ya que contiene gran parte de los componentes de un computador convencional pero a un menor precio. En este proyecto se desea trabajar con una Raspberry Pi 3, por lo tanto se presentará un poco de su historia, la evolución que ha tenido al paso del tiempo y las distintas alternativas que se encuentra en el mercado con el objetivo de obtener mejoras en la implementación del sistema desarrollado.

Es importante señalar que la Raspberry comenzó su fabricación en el Reino Unido por la fundación Raspberry Pi, con fines de estimular la enseñanza de la ciencia de computación en las escuelas.

Existen distintos modelos de Raspberry Pi, en total son 8 los cuales se mencionarán más adelante, pero en este caso se trabajará con el modelo 3B, ya que es el modelo más avanzado fabricado hasta ahora.

Además se expondrán distintas aplicaciones que ya existen en el mercado y que se han implementado en este tipo de placas, con el propósito de conocer su comportamiento con la librería OpenCV y su rendimiento bajo estas condiciones.

### 3.1 Introducción a la RPI

Este producto es de propiedad registrada pero de libre uso, por lo tanto lo que se realice en ella queda a libre disposición. Es tal su masificación y efectividad que ha sido utilizada por empresas para el desarrollo de sus productos y con el paso del tiempo se ha convertido en una mejor opción que las ya conocidas placas Arduino.

Existe un sin número de implementaciones a las que puede utilizarse una RPI, tanto para el control de un robot, domótica, proyector de videos, procesamiento digital de imágenes, etc. Además es de fácil acceso ya que en la Web se pueden encontrar múltiples distribuidores de ellas.

Cuenta con un sistema operativo oficial llamada RaspBian, la cual es una versión adaptada de Debian, pero al ser una placa Open Source, es posible implementar otros sistemas operativos, según desee el operador. Por otra parte es posible agregar distintos accesorios que permiten

realizar implementaciones con un mejor desempeño, por ejemplo, cuenta con una cámara de 14 megapíxeles, la cual es posible conectarla en un compartimiento ideado para ella [22].

Un ejemplo de la gran capacidad que poseen estas placas ha sido la de desarrollar una impresora 3D (Figura 3-1), de modo que la RPI primero escanea el objeto en tres dimensiones y luego se procede a la impresión [23].

### 3.2 Características generales

La Raspberry Pi 3 (Figura 3-2) aparece en el mercado al año siguiente de la aparición de su antecesora, la RPI 2, la cual se lanzó el año 2015, entre una de sus principales características es que su precio no varía mucho, por lo que hace que la mayoría de las personas puedan acceder a ella, cuesta \$38.765 aproximadamente, contra los \$37.221 aproximados de su predecesora. Otro de sus beneficios es que no es complicada de aprender a utilizar, ya que existen diversos tutoriales en la web para realizar distintos tipos de proyectos y con distintos niveles de dificultad.

Entre sus características técnicas es posible mencionar las más destacadas, en donde una de las principales ventajas se encuentra en su procesador ARM Cortex A53, un procesador de 4 núcleos a 1.2 GHz de 64 bits, por lo tanto, es posible señalar que su rendimiento es de unas 10 veces superior a la RPI original y un 50% más que su antecesora. Además cuenta con Bluetooth 4.1 y WiFi integrados, cosa que las versiones anteriores no poseían, por lo que solo se podían realizar conexiones mediante puerto USB o mediante conexión Ethernet, lo que permite ampliar su utilización en una mayor cantidad de proyectos.

Otro de los beneficios que se pueden encontrar en esta nueva versión, es que a pesar de los de las características añadidas, sus dimensiones físicas se mantienen al igual que la forma, en comparación a la RPI 2. Por su puesto, las características básicas encontradas en las versiones anteriores se mantienen, vale decir que se mantiene la ranura para la tarjeta de memoria microSD, los 4 puertos USB, la conexión mediante puerto Ethernet y Full HDMI, interfaz de cámara y de pantalla, conector de audio y video RCA, entre otras cosas. Cabe señalar que también cuenta con una cámara, la RaspiCam, la cual contiene una sección específica dentro de la placa, por lo que solo es instalarla y programarla, el detalle está en que se vende como un componente por separado [35].



Figura 3-1: Pieza realizada en una impresora controlada por RPI (fuente: <http://iniciativa3d.blogspot.cl>).



Figura 3-2: Raspberry Pi 3 (fuente: <http://raspi.tv>).

### 3.3 Lenguaje de programación en RPI

Por defecto la RPI viene con el lenguaje de programación Python, por ello su nombre de “Pi”, por este motivo se recomienda programar en este tipo de lenguaje en la placa, esto no implica que no se puedan utilizar otro tipo de lenguaje. Para la utilización del lenguaje C++, por ejemplo, no es necesario realizar alguna instalación adicional de algún software que permita programar en dicho lenguaje, salvo que para ello la programación se realiza a través de la consola desplegada en pantalla. Una forma más amigable para programar en C++ es utilizando un editor de texto que permita realizar dicha programación y que luego genere un archivo con la extensión correspondiente a este lenguaje.

Como se ha mencionado anteriormente, es posible programar en cualquier tipo de lenguaje la condición es poseer una editor de texto acorde al lenguaje utilizado y la otra opción es la instalación de un compilador que soporte dicho lenguaje.

### 3.4 Aplicaciones realizadas

Existe un sin número de aplicaciones que se pueden realizar con este computador de placa reducida, pero el objetivo de este proyecto consiste en la detección de personas, por lo que se presentarán proyectos que tiene relación con este tipo de aplicaciones.

Uno de los principales proyectos que se pueden encontrar en cuanto a la detección de un objeto, tiene que ver con la detección de rostros, para ello se recomienda utilizar alguna librería que permita realizar este tipo de procesamiento. En este caso, y como se mencionó anteriormente, se decidió utilizar OpenCV, por lo que se buscó el método de incorporar esta librería en la RPI. Otras de las aplicaciones que se pueden encontrar y en donde se ha utilizado este tipo de placas, corresponde a la detección de objetos, como también a al seguimiento de objetos producto de un color determinado.

La Figura 3-3 corresponde a un sistema de detección y contador de personas creado a partir de una RPI 1, por lo que se necesitaron componentes adicionales para realizar dicho programa, por ejemplo se utilizó un detector de movimiento para realizar el conteo y un adaptador de WiFi con puerto USB, para compartir la información de manera instantánea.

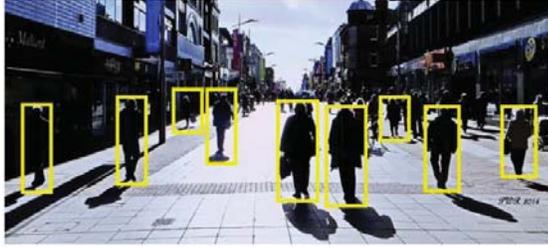


Figura 3-3: Contador de personas en RPI (fuente: <https://ubidots.com>).

Otras de las aplicaciones más comunes que se pueden encontrar en cuanto al reconocimiento de objetos, tiene que ver con la detección de rostros. Son muchas las aplicaciones que se pueden encontrar en la web con este tipo de ejemplos, en la Figura 3-4 se muestra un ejemplo de ello.

Las aplicaciones mencionadas anteriormente, llevan bastante tiempo desde que han sido desarrolladas, por lo que los componentes utilizados para su implementación, no son los mismos. Por ejemplo la placa RPI ya que se ha ido actualizando rápidamente, por lo tanto, esto hace pensar que una posible aplicación de detección y conteo de personas, puede ser efectivo. Además el constante mejoramiento de estas placas, permite desarrollar implementaciones capaces de realizar visión por computador, área que cada vez tiene mayores adaptaciones en la industria.



Figura 3-4: Contador de personas en RPI (fuente: <https://ubidots.com>).

### 3.5 Modelos de RPI

Como ya se mencionó anteriormente, existen distintos modelos de RPI en el mercado, pero la última versión desarrollada corresponde a la versión 3B, además de ser la placa con la que se desea implementar el descriptor HOG.

En este apartado se realizará un pequeño cuadro comparativo entre los distintos modelos de RPI (Tabla 3-1) que han salido al mercado, en donde se considerarán los aspectos más relevantes que poseen estas placas, con el objetivo de señalar el porqué de la decisión de utilizar el modelo 3B, en desmedro de los otros.

Tabla 3-1: Modelos de RPI [16].

Raspberry	Modelo I		Modelo +		Modelo B	
	RPI A	RPI B	RPI A+	RPI B+	RPI 2B	RPI 3B
CPU	700 MHz	700 MHz	700 MHz	700	900 GHz	1.2 GHz
RAM	256 MB	512 MB	256 MB	512 MB	1 GB	1 GB
Puerto USB	1	2	1	4	4	4
Ethernet	No	Sí	Sí	Sí	Sí	Sí
Dimensiones (mm)	54x85	54x85	58x56	85x56	85x56	85x56
Alimentación	5V	5V	5V	5V	5V	5V

Además existen otras alternativas de placas reducidas, las cuales podrían presentar mejor rendimiento debido a las características que estas presentan, esto es en caso de que la RPI 3 no cumpla con las expectativas previas a su funcionamiento. Por lo tanto es necesario contar con otras posibles sustituciones a esta placa.

En esta ocasión se compararán 3 placas distintas (Tabla 3-2), con distintos procesadores, capacidad de memoria RAM, sistema operativo compatible y precio en el mercado. Esto porque son placas que pueden cumplir la misma función de realizar procesamiento digital de imágenes, y su respectiva aplicación en visión por computador.

Es necesario señalar que la decisión de utilizar una RPI 3 para la realización del proyecto, tiene relación a que dentro de los distintos modelos existente de Raspberry, el modelo 3B es el que posee mejores características. A su vez, las alternativas de placas UDOO poseen un elevado costo comercial, por lo que se hace difícil adquirir una de ella. Por lo tanto, considerando precio y características técnicas, hace que la mejor alternativa para ser utilizada, sea la correspondiente a la de una placa RPI 3.

Tabla 3-2: Comparación entre placas.

Placas Reducidas	RPI 3	UDOO ADVANCED	UDOO ULTRA
Procesador	ARMv8CPU	Intel	Intel N3710
RAM	1 GB	2.24 GB	2.56 GB
Sist. Operativo*	L/A/W	L/A/W	L/A/W
Precio**	40.958	102.224	199.965

\*L = Linux, A = Android, W=Windows 10.

\*\*Valor del dólar \$636 chilenos, al día 20-05-2018.

## 4 Construcción del sistema

Mediante el descriptor HOG, se desea implementar un sistema que permite detectar personas en tiempo real, además de seguir su trayectoria a lo largo de la escena, con el objetivo de conocer el flujo de gente que se puede presentar en un determinado lugar. Este sistema debe ser adaptado a una Raspberry Pi 3, por lo que se requiere de una serie de pruebas para lograr un óptimo funcionamiento del detector.

En primer lugar se realizarán pruebas sobre imágenes en donde se encuentre presencia de personas, con el objetivo de conocer el comportamiento del detector en la RPI 3. A su vez, se realizará una comparación frente a un computador de escritorio, de manera de evaluar las detecciones efectivas de personas y el tiempo que demora en la ejecución del sistema.

En caso de ser exitosa la primera parte de las pruebas, se procederá a realizar detecciones en videos descargados desde internet, para saber si el sistema es capaz de detectar personas en movimiento y realizar su respectivo seguimiento de forma constante. Esto con el objetivo de, posteriormente, aplicar el detector en tiempo real, ya que los factores externos podrían tener mayor influencia al momento de efectuar una clasificación.

### 4.1 Trabajo previo

Lo primero que debió realizarse para la utilización de la RPI 3 corresponde a la configuración de la misma instalando en primer lugar el sistema operativo Raspbian. Para ello es necesario contar con una memoria tipo tarjeta microSD, se recomienda utilizar una con capacidad mayor o igual a 16 GB libres de almacenamiento, para no tener problemas de espacio de memoria. Luego fue necesario instalar las actualizaciones correspondientes, más la configuración básicas necesaria y los programas requeridos para la inicialización del detector de personas. El siguiente paso corresponde a la instalación la librería OpenCV, en este caso se descargó la versión 3.1 [19] para el trabajo tanto de prueba en el computador, como también en la placa, ya que con las distintas versiones, las funciones al momento de la programación varía entre una y otra. Además se decidió instalar un compilador que permitiera crear archivos en lenguaje C++, ya que se ha decidido utilizar este tipo de lenguaje debido a las ventajas que posee por su rapidez en la ejecución de las tareas.

También es necesario señalar que la cámara utilizada para la captura de las imágenes, es una cámara con puerto USB Logitech c270, que se encuentra disponible en el laboratorio de robótica.

## 4.2 Estructura del descriptor

A continuación se expondrán los parámetros que permiten configurar el sistema para que sea capaz de realizar detección de personas, y como se mencionó anteriormente, se utilizarán las herramientas disponibles en OpenCV.

Lo primero que se debe realizar es definir el lenguaje con el que se trabajará para escribir el programa, en este caso se optó por C++, además de la utilización de las librerías correspondientes para los distintos procesamientos a las imágenes.

La Tabla 4-1 presenta las funciones principales utilizadas para la creación del sistema que permite detectar personas:

Cabe señalar que la función `detecMultiScale` necesita definir ciertos valores en algunos parámetros que permiten obtener mejores detecciones. La Tabla 4-2 muestra estos parámetros y su respectiva descripción [94]. Además, según el equipo que se esté utilizando, ya sea computador o RPI, es posible modificar estos valores según las características del equipo, para así lograr el máximo de rendimiento posible.

Tabla 4-1: Funciones utilizadas para efectuar detección de personas.

Función	Descripción
<b>HOGDescriptor</b>	Especificación que se están realizando detecciones por medio del histograma de gradientes orientadas.
<b>setSVMDescriptor</b>	Para la clasificación de los objetos analizados se utilizará la máquina de vectores de soporte.
<b>getDefaultPeopleDetector</b>	Se define que el objeto a detectar corresponde a personas.
<b>detecMultiScale</b>	Definición de valores para distintos parámetros utilizados para una correcta clasificación de personas.

Tabla 4-2: Parámetros ajustables que permiten la detección de personas.

Parámetro	Descripción
<b>img</b>	Imagen de entrada a analizar.
<b>found_locations</b>	Límites para los objetos detectados.
<b>hit_threshold</b>	Distancia entre las clases de entrenamiento, con el hiperplano de SVM.
<b>win_stride</b>	Ventana de deslizamiento. Debe ser múltiplo de los bloques (2x2).
<b>padding</b>	Parámetro que permite mantener la compatibilidad con la CPU. Debe ser (0,0).
<b>scale0</b>	Coefficiente de incremento en la ventana de detección (Píxeles de relleno).
<b>group_threshold</b>	Umbral de similitud. Cuando se detecta un objeto, puede ser cubierto por muchos rectángulos, se aconseja que su valor sea 2.

Finalmente se procede a realizar un conteo de las personas detectadas, en este caso lo que se propone es contabilizar la cantidad de rectángulos presente en la escena. Por lo tanto no se realiza un contador independiente al detector, sino que a medida que se detecta una persona, esta es inmediatamente contabilizada.

#### 4.2.1 Estructura del sistema

En esta sección se explicará de manera general, la estructura que posee el detector de personas. La Figura 4-1 muestra un diagrama de bloques que explica el proceso, y la Figura 4-2 muestra algunas imágenes obtenidas con el sistema propuesto.

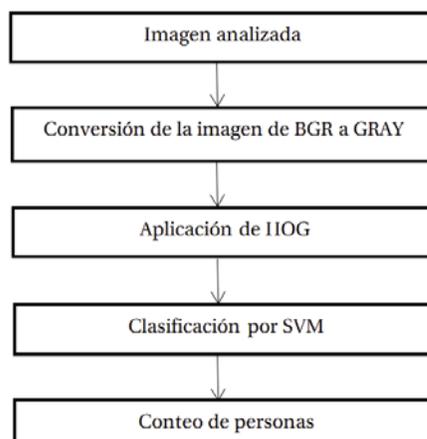


Figura 4-1: Diagrama de bloques del sistema desarrollado.

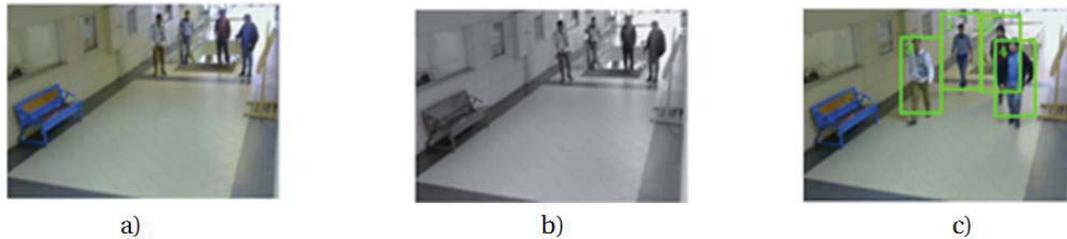


Figura 4-2: Sistema propuesto, donde: a) imagen analizada, b) conversión de imagen de BGR a GRAY, c) Aplicación del descriptor para detección y conteo de persona.

### 4.3 Análisis de HOG en RPI 3 y Computador

El objetivo de este análisis, es conocer el comportamiento del detector de personas en la RPI 3, para conocer su efectividad en las detecciones y el tiempo que demora en realizar dichas trabajo. Además se realizarán comparaciones con un computador, de esta forma, determinar si es conveniente seguir con la utilización de esta placa para la continuidad del proyecto.

#### 4.3.1 Experimentación en imágenes

Esta experimentación corresponde a la primera prueba realizada en la RPI 3, que permitirá determinar si es posible ejecutar el detector de personas en ella, para ello se compararán resultados con un computador para determinar si el funcionamiento con la RPI es adecuado para seguir trabajando en ella. En la Tabla 4-3 se especifican las características técnicas por parte del computador utilizado.

En primera instancia se procedió a analizar 50 imágenes con un total de 86 personas, la Figura 4-3 es un ejemplo de las imágenes utilizadas, y en el cual se obtendrá el nivel de efectividad para la detección de ellas. En la Tabla 4-4 se muestran los resultados obtenidos en esta prueba. Cabe destacar que el tiempo que demora el sistema en efectuar las detecciones, es el mismo para los dos casos estudiados, tardándose menos de 10 segundos en realizar una detección, por lo que no es necesario mencionarlo en la tabla comparativa.

El obtener más de un 70% de efectividad en la detección de personas, ha permitiendo dar paso a la ejecución del descriptor en un video. De esta forma se realizará el mismo cuadro comparativo para determinar si la alta efectividad lograda en imágenes, es posible mantenerla en un video.

Tabla 4-3: Características del computador utilizado.

Características del computador	HP Pavilion g4
Procesador	AMD Dual-Core E-450
RAM	4 GB
Disco duro	500 GB
Sistema operativo	Ubuntu 16.04

Tabla 4-4: Análisis del HOG en fotografías.

Detección de personas	Experimento I	
	Computador	RPI 3
Total de personas	86	86
Total detectado	63	63
Falsos Positivos	2	2
Efectividad	73,26%	73,26%



Figura 4-3: Detección de personas en fotografía en RPI 3.

### 4.3.2 Experimentación en video

El siguiente análisis corresponde a la ejecución del detector de personas en un video cualquiera (Figura 4-4), en el cual se puede apreciar un flujo constante de personas. Lo que se espera en esta experimentación, tiene relación con una detección continua de las personas a lo largo de su recorrido, por lo que de ser efectivas las detecciones, se realizaran pruebas en tiempo real.

Al igual que en el caso anterior, se compararán los resultados obtenidos tanto en la RPI 3, como en computador. Para ello se presentan los resultados en la Tabla 4-5.

Es necesario precisar que el video fue descargado desde internet, en donde prevaleció la buena calidad de éste y en donde se pudiera apreciar claramente el tráfico de personas, además de que estuviera libre de publicidad o letras, ya que podría influir al momento de la clasificación.

La duración del video escogido para la experimentación es de 12 segundos, esto permite determinar el tiempo de ejecución tras aplicarle el detector de personas al video, ya que este tipo de procesamientos aplicados, podría significar un aumento en el tiempo de reproducción.

En esta ocasión la información recolectada no permite realizar conclusiones relevantes, ya que al obtener un elevado tiempo de ejecución del video (en computador cercano a los 4 minutos y en la RPI más de 1 hora), hace que se pierda validez en los resultados, puesto que no se desea implementar un sistema que tarda 300 veces más, en la reproducción de algún video.

Sin embargo, se ha dispuesto realizar pruebas en tiempo real, para saber si el comportamiento del detector es similar el observado en un video, o por el contrario, tiene mejor rendimiento.

Tabla 4-5: Detección de personas en video.

Detección de personas	Experimento II	
	Computador	RPI 3
Total de personas	56	56
Total detectado	33	49
Falsos Positivos	0	0
Efectividad	58,93%	87,5%
Tiempo ejecución	3'56''	60'57''

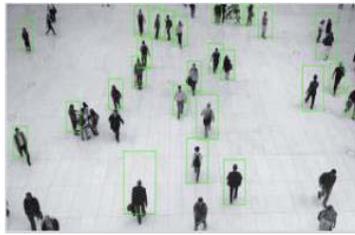


Figura 4-4: Detección de personas en video en RPI 3.

### 4.3.3 Experimentación en tiempo real

Finalmente se procederá a realizar detecciones en tiempo real (Figura 4-5) por parte del detector de personas, ya que los resultados obtenidos en el experimento anterior no son buenos en cuanto el tiempo de ejecución del video, por lo tanto, se espera conseguir un mejor rendimiento por parte de la RPI 3, puesto que se desea lograr una gran cantidad de detecciones junto con un bajo tiempo de ejecución.

Debido a que es posible realizar ciertos ajustes tanto en la calidad de la imagen obtenida, como también la cantidad de frame definidos para la reproducción de la cámara, se espera reducir la cantidad de recursos utilizados para la ejecución del sistema.

En esta ocasión se realizaron tres pruebas distintas, en donde solamente se utilizó la RPI 3, ya que se logró determinar que si bien, es posible implementar este tipo de sistemas en uno de estos dispositivos, la dificultad se centra en bajar los tiempos de ejecución del mismo, por lo que ya no es relevante la utilización de un computador, puesto que no es un objetivo del proyecto realizar comparaciones entre estos dispositivos.

La primera prueba se efectuó con el objetivo de analizar el comportamiento del sistema en cuanto al tiempo de ejecución del mismo y el retardo presente al momento analizar el video en tiempo real, por lo que solo se utilizó a una sola persona en un trayecto determinado. Con la segunda prueba se somete al sistema a un mayor flujo de personas, cercana a las 20, para así conocer el nivel de efectividad por parte del detector. Por último, en la tercera prueba se espera identificar los elementos que ayudan a la detección, como también los que afectan al buen desempeño del mismo, tanto externos como a los correspondientes a la configuración de los distintos parámetros utilizados.

A continuación se presenta la Tabla 4-6, en donde se registra los resultados obtenidos en estos primeros experimentos realizados.

Tabla 4-6: Detección de personas en tiempo real en RPI 3.

Detección de personas	Prueba I	Prueba II	Prueba III
Personas	1	18	128
Detecciones	1	13	94
Falsos positivos	0	1	18
Efectividad	100%	72.22%	73.44%

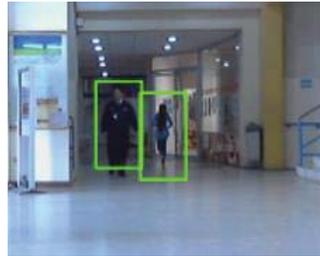


Figura 4-5: Detección de personas en tiempo real en RPI 3.

El objetivo principal de realizar pruebas en tiempo real tiene relación a que los resultados obtenidos en computador fueron deficientes, ya que no cumplen con las expectativas deseadas; lograr una gran cantidad de detecciones y un bajo tiempo de ejecución. Por esta razón es que, al poder manejar cierto tipo de variables, las que se encuentran directamente relacionadas con la rapidez en el análisis por parte del detector.

#### 4.4 Análisis de lo realizado

En este capítulo se presentaron las primeras pruebas realizadas para la detección de personas mediante el descriptor HOG, en primera instancia se pretendía comprobar su funcionamiento analizando imágenes y videos, tanto en computador como en la RPI 3. Un punto a destacar, es que se lograron muy buenos resultados en la detección de personas en una imagen, pero al analizar un video, el rendimiento por parte de los dos equipos utilizados fue muy deficiente, impidiendo realizar observaciones importantes sobre el comportamiento del sistema, motivo por el que se decidió realizar pruebas en tiempo real.

**Respecto a los 2 sistemas propuestos:** El objetivo de realizar pruebas en dos equipos tiene relación con la comprobación, en primera instancia, del funcionamiento del sistema propuesto tanto es su estructura, como en su desempeño en la RPI 3, ya que no se tenía certeza sobre su implementación en este tipo de dispositivos, por lo tanto se hacía necesario contar con un computador para realizar una comparación.

Luego de comprobar que funcionamiento de OpenCV en ambos equipos, se procede a realizar algunos análisis a imágenes con presencia de personas, luego en videos y finalmente en tiempo real. Los rendimientos presentados por ambos computadores en las primeras pruebas fueron

muy similares, ya que la utilización de recursos por parte de estos no era demasiado alto, pero al analizar imágenes en movimiento, el nivel de efectividad se ve disminuido ya que se requiere de mejores características en cuanto a la velocidad de procesamiento en los equipos.

**Respecto al tiempo de ejecución:** En primera instancia los tiempos de ejecución entre un equipo y otro fueron similares al momento de analizar imágenes, permitiendo seguir avanzando con los análisis a mayor escala. En ese sentido al procesar videos, el computador logra su reproducción en mucho menor tiempo en comparación a la RPI 3, pero su nivel de efectividad en las detecciones también se ve disminuido, esto se debe a que si bien la reproducción del video fue más rápida, se pierde información entre los frame de reproducción, por lo que no se logra detectar a una persona mediante su análisis por HOG, cuando ya ha cambiado de postura y posición. Por el contrario, lo que ocurre en la RPI es que, a pesar de ser mucho más lenta la reproducción del video, no se pierde parte del mismo entre los frame reproducidos, permitiendo realizar las detecciones de mejor manera. Por lo tanto, una rápida reproducción en un video, no implica una mejor efectividad en las detecciones, mientras que una lenta reproducción, significa una mayor cantidad de detecciones correctas, pero a un alto costo de tiempo tardando en ejecución del mismo, motivo por el cual se decidió realizar pruebas en tiempo real.

**Respecto a las pruebas en tiempo real:** El objetivo de realizar este tipo de pruebas tiene relación, en primera instancia, en conocer el comportamiento del sistema, ya que en caso de ser similar a lo ocurrido en video, implicaría analizar otras alternativas de computadores de placa reducida para la continuidad del proyecto.

Por otra parte, al realizar pruebas en tiempo real, permite efectuar ajustes en cuanto a la calidad de la imagen obtenida, tanto en la resolución de la cámara, como la cantidad de frame utilizados para la obtención del video. Esto permite disminuir de forma considerable la cantidad de recursos utilizados por la RPI para la ejecución del programa, por lo que el tiempo de retardo producido entre la captura de la imagen y su procesamiento, es mucho menor a lo presentado en el análisis de video.

**Respecto al rendimiento:** En primera instancia, el rendimiento por parte de los dos equipos con los que se trabajó, fueron similares y muy efectivos, pero al momento de realizar pruebas en video la situación cambió, esto se debe principalmente a que la RPI no posee las mismas características a las del computador, por lo que era de esperarse lo ocurrido. No obstante, el nivel de detección sí fue significativo entre uno y otro, lo que provoca una complicada situación; se prefiere rapidez en la ejecución del video, afectando directamente a las detecciones, o predomina una alta tasa de aciertos, pero con un elevado tiempo de reproducción del sistema.

En cuanto a análisis en tiempo real solo se obtuvo información sobre la efectividad en las detecciones por parte de la RPI 3, ya que no se hizo necesario seguir utilizando un computador como base comparativa, puesto que no es parte del proyecto realizar este tipo de comparación, además, al lograr implementar el detector de personas en la Raspberry, predominan los estudios en ella, ya que es la base del proyecto en cuestión.

## 5 Experimentación

En este último capítulo se realizarán pruebas en situaciones reales, donde se pretende obtener el mejor rendimiento del detector de personas, por lo tanto se debe especificar todas las características relevantes que permitieron definir la configuración del sistema propuesto, tanto en el ajuste del sistema, como en los factores externos que pueden afectar la detección.

En primer lugar se realizarán pruebas con un número determinado de personas, y bajo situaciones controladas, esto hace referencia a la disposición en la que circulan por el sector a realizar el experimento, de manera de determinar el sector donde se efectúa detección. De esta manera conocer los posibles problemas que se puedan presentar, al momento de realizar pruebas bajo condiciones de flujo de personas no controladas.

Posteriormente se realizarán las pruebas en condiciones que el flujo de personas, y su trayectoria, no pueden ser controlados. El objetivo de esto es poder verificar el funcionamiento del programa ejecutado, validando su efectividad, y determinando bajo qué condiciones este sistema logra su mejor rendimiento.

Se espera que el nivel de efectividad en la detección de personas, sea de a lo menos el 90% para ser considera un sistema confiable, además de obtener la mínima cantidad de falsos positivos, idealmente se espera que esta sea cero, en caso contrario, se espera que este número sea el mínimo posible.

### 5.1 Ajustes de parámetros extrínsecos

Con este ajuste de parámetros, se definen los valores en las variables que permiten realizar mejores detecciones, como lo es el clasificador por SVM, donde se determina la proximidad de las clases (personas, no personas) con el hiperplano.

También es posible definir el tamaño de las imágenes utilizadas para el entrenamiento, tal como recomienda Dalal y Triggs [16], la imagen debe ser de 64x128 píxeles, pero este puede variar si se desea, por lo que se presume, podría permitir realizar detecciones a personas que se encuentran muy distantes al lugar de la prueba.

La siguiente tabla (Tabla 5-1) muestra a nivel general los valores de las configuraciones efectuadas, con el objetivo de definir con cuál continuar trabajando.

Tabla 5-1: Niveles de detección, ajustando parámetros extrínsecos.

Detección de personas	SVM		Tamaño de imagen	
	0	1	48x96	64x128
Detecciones	Alto	Alto	Medio	Alto
Falsos Positivos	Bajo	Medio	Bajo	Medio
Falsos Negativos	Bajo	Medio	Medio	Bajo

## 5.2 Ajuste de resolución de video

En este tipo de ajuste, se espera definir el tamaño de resolución con los que se trabaja en la cámara, ya que es un factor muy importante al momento de realizar las detecciones, debido a que no solo permite una mejor clasificación de las personas, sino que también tiene directa relación con el retardo que se produce en la ejecución del programa.

Además se logra identificar claramente un área de detección (Figura 5-1), en donde cada vez que transita una persona por aquella área, la probabilidad de ser detectada es considerablemente alta.

En la siguiente tabla (Tabla 5-2) se logra obtener información sobre la efectividad en las detecciones, dependiendo de la resolución utilizada.

Tabla 5-2: Niveles de detección según resolución de la cámara.

Detección de personas	Resolución		
	176x144	352x288	640x480
Detección	Bajo	Alto	Medio
Falsos Positivos	Bajo	Bajo	Bajo
Falsos Negativos	Bajo	Bajo	Medio
Retardo	0,5''	3''	12''



Figura 5-1: Área de detección a 270 [cm] de altura.

### 5.3 Ajustes de posición de la cámara

Un factor que a medida que se realizaron las pruebas, tuvo gran importancia, tiene relación con la altura en donde se posiciona la cámara, ya que se logró identificar un área de detección, en donde se puede aplicar de mejor manera la clasificación de personas.

Si bien se realizaron una mayor cantidad de pruebas, solo se detallarán dos (Tabla 5-3), que es en donde el detector obtuvo su mejor desempeño.

La Figura 5-2 y Figura 5-3 muestran la disposición de la cámara en las pruebas realizadas en el tercer piso de la facultad de ingeniería de la PUCV.

Tabla 5-3: Nivel de detección según altura de la cámara.

Detección de personas	Altura	
	110 cm	270 cm
Detección	Medio	Alto
Falsos Positivos	Bajo	Bajo
Falsos Negativos	Bajo	Bajo



Figura 5-2: Posición de la cámara a 270 [cm] y el área no visible.



Figura 5-3: Posición de la cámara a 110 [cm] y el área no visible.

## 5.4 Experimento con flujo de personas controladas

Este tipo de experimento consiste en definir un número determinado de personas que transitará por un lugar específico (Figura 5-4), de manera de conocer el comportamiento del sistema realizado, con el objetivo de validar su nivel de detecciones. Además, se debe determinar el cómo transitarán estas personas sobre el área de interés.

En ese sentido se cuenta con 5 personas, las cuales realizarán su paso por el lugar establecido en distintas configuraciones, comenzando con una separación importante entre uno y otro, hasta terminar lo más próximo posible entre ellos. Se determinó realizar 6 vueltas por el sector, 3 de frente a la posición de la cámara y 3 de espalda a la cámara, para conocer si existe alguna variación en cuanto a las detecciones frontales y posteriores de una persona.

En primer lugar fue necesario definir el tamaño de la resolución que se utilizará para realizar las pruebas, en este caso, y como se analizó anteriormente, la resolución con mayor cantidad de aciertos, junto con disponer una mayor área de detección, corresponde a la configuración de 352x288 píxeles. Por lo tanto esta, y las siguientes pruebas a analizar, se determinará el uso de este tipo de resolución.

Por otra parte, se determina situar al cámara a 270 cm de altura, ya que a esa elevación el nivel de detección es mayor, por lo que para las siguientes experimentaciones, se utilizará esta altura recomendada.

La tabla (Tabla 5-4) muestra los resultados obtenidos para esta prueba, tanto para la cantidad de detecciones realizadas, la cantidad de falsos positivos obtenidos y el porcentaje de acierto.

Tabla 5-4: Pruebas en RPI 3 con flujo controlado de personas.

Detección de personas	Prueba
	352x288
Total de personas	31
Detecciones	29
Falsos positivos	1
Efectividad	93,55%
Retardo de video	3''



Figura 5-4: Prueba con flujo controlado de personas.

## 5.5 Experimento con flujo de personas no controlado

En el siguiente experimento, se evaluarán siete situaciones en donde el sistema debe realizar la detección de personas bajo condiciones que no fueron programadas, por lo que el comportamiento de los peatones no debe seguir un patrón definido. Cuatro de estas pruebas son de corta duración, no más de 10 minutos, para conocer el comportamiento del detector, y las siguientes tres pruebas son de larga duración, la más extensa es de alrededor de 18 horas. En ese sentido, las cuatro primeras situaciones analizadas son completamente distintas unas de otras, mientras que las tres siguientes son realizadas en el mismo lugar, tercer piso de la facultad de ingeniería de la PUCV. El objetivo principal de estas pruebas, es obtener la mayor cantidad de detecciones correctas, junto con conocer las limitaciones que posee el sistema que afectan directamente a la clasificación de personas.

Como se mencionó en la sección anterior, se decidió utilizar la resolución de 352x288 píxeles ya que fue la configuración que mejores resultados presentó a lo largo de las pruebas controladas, siempre teniendo en consideración que el retardo presentado en la proyección del video será de 3 segundos. En cuanto a la altura en que se dispondrá la cámara, difiere a las pruebas realizadas anteriormente, ya que en este caso, hay que adecuarse a las características del lugar en donde se realizarán estas pruebas, por lo que los 2,7 metros no siempre se podrán cumplir.

A continuación se detallarán las cuatro primeras pruebas efectuadas para la detección de personas y sus resultados se pueden observar en la Tabla 5-5:

- I) La primera prueba se realizó en la entrada de la estación de metro Viña del Mar (Figura 5-5a), en donde se espera que los resultados sean muy favorables, ya que se trata de una estación subterránea y como se mencionó anteriormente, el sistema se comporta de buena forma al realizar detecciones en altura. Por otra parte, al ser una de las estaciones de metro con mayor tránsito de personas, se espera que puedan obtenerse resultados muy significativos para el comportamiento del detector.
- II) La segunda prueba se efectuó en uno de los semáforos ubicados en la plaza Viña del Mar (Figura 5-5b), en donde el flujo de personas es mucho mayor de lo esperado, ya que se trata de uno de los sectores más transitados de la ciudad. Por lo que el comportamiento del sistema podría arrojar resultados importantes en cuanto a su verdadero funcionamiento, ya que se trata de una de las pruebas con mayor flujo de gente registrado hasta ahora. Cabe señalar que la posición de la cámara en esta ocasión es de mediana altura, no supera los 1,5 metros de elevación, por lo que se puede ver afectado el verdadero nivel de detección por parte del sistema.
- III) La tercera prueba fue realizada en el bandejón central de Avenida Brasil, Valparaíso (Figura 5-5c) frente al edificio anteriormente mencionado (FIN-PUCV), en donde si bien no se presentó un tráfico significativo de personas, fue posible obtener conclusiones importantes en cuanto al funcionamiento del sistema en condiciones externas presentes en la escena. Para ello se utilizaron las estructuras disponibles en

el sector, las cuales permitieron realizar esta prueba con la especificación deseada de situar la cámara a una altura considerable.

Las siguientes tres pruebas se realizaron desde el laboratorio de robótica de la EIE, mismo lugar donde antes se habían realizado los trabajos para la configuración del sistema. En esta ocasión, las pruebas fueron de larga duración, para obtener un análisis más global respecto al comportamiento del detecto. El detalle de estas pruebas se mencionarán a continuación y sus resultados se encuentran en la Tabla 5-5:

- IV) La cuarta prueba fue realizada a la altura ideal señalada, 2,7 metros de altura, en esta oportunidad se analiza a un grupo importante de personas, que se encuentran reunidos en un espacio reducido. Por lo tanto, el sistema será sometido a una situación que se presume debe ser favorable, puesto que uno de los motivos principales para utilizar la cámara en altura, tiene relación a este tipo de problemática, el lograr diferenciar a una persona de otra, al encontrarse muy próximos entre sí.
- V) La quinta prueba, a diferencia de la prueba anterior, se realizó a los 1,1 metros de altura, ya que se dispuso en otro sector del laboratorio, situación similar a la que se presentó en la prueba III, por lo que se podrán obtener conclusiones relevantes al respecto, ya que si presentan alguna variación importante, en comparación a la prueba realizada anteriormente y bajo condiciones similares, se debe señalar el motivo de aquello.
- VI) La sexta y última prueba, vuelve a realizarse en las mejores condiciones que la quinta prueba, la diferencia se encuentra en que en esta ocasión se realizará una prueba de 18 horas continuas de análisis. De esta forma será posible conocer el comportamiento del detector, por ejemplo de noche, si en algún momento la RPI 3 falla producto de lo extenso de la grabación, entre otros aspectos.

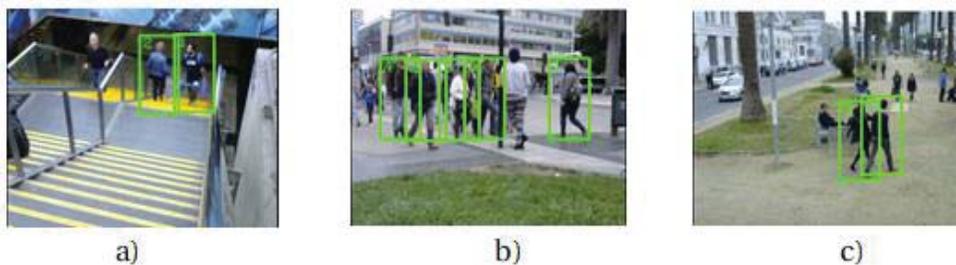


Figura 5-5: Experimento con flujo no controlado de personas. a) Prueba I; Estación de metro, b) Prueba II; semáforo, c) Prueba III; Avenida Brasil.

Tabla 5-5: Pruebas en RPI 3 con flujo no controlado de personas.

Detección de personas	Prueba I	Prueba II	Prueba III	Prueba IV	Prueba V	Prueba VI
Personas	7	42	21	94	158	780
Detecciones	6	28	17	88	121	724
Falsos positivos	4	6	0	18	12	29
Efectividad	85,71%	66,67%	80,95%	93,62%	65,41%	92,82%

## 5.6 Análisis de los resultados

Las principales conclusiones que se lograron obtener, luego de realizado los distintos tipos de experimentos y ajustes sobre el sistema, se pueden separar en dos partes. La primera tiene relación con los resultados analizados al realizar un control sobre el flujo de personas, esto significó conocer las limitaciones propias del detector, junto con las características propias de cada una de las configuraciones, permitiendo efectuar algunas modificaciones con las que se pretende obtener la mayor cantidad de aciertos de personas detectadas.

De esta forma, se utilizaron distintas configuraciones sobre algunos parámetros específicos, como lo ocurrido con la SVM, el tamaño de la imagen utilizada para entrenar dicho clasificador, la resolución de la cámara y la altura desde donde se sitúa la misma para realizar las detecciones, permitiendo aumentar la cantidad de detecciones efectivas de las personas.

La segunda parte, tiene relación con los resultados obtenidos al momento de realizar pruebas con un flujo de personas que no son controladas, permitiendo observar el comportamiento del sistema desarrollado en situaciones reales, siendo necesario identificar las falencias presentes, ya sea producto del desarrollo del detector, como también debido a los factores externos presentes en el lugar de las pruebas, como la construcción, iluminación, objetos, entre otros, siendo factores directos que afecten la detección.

**Respecto a los ajustes de parámetros:** se pudo determinar que modificando ciertos valores en los parámetros mencionados anteriormente, el detector logra mayores niveles de detección. En donde por ejemplo, en la SVM, mientras más cercana se encuentren las clases utilizadas al hiperplano, más certeras son las detecciones, impidiendo de esta forma obtener una gran cantidad de falsos positivos. En cuanto al tamaño se las imágenes utilizadas para el entrenamiento del mismo clasificador, se determinó que la mejor resolución corresponde a las de 64x128, ya que la otra configuración (48x96 píxeles) también provoca un aumento en la tasa de falsos positivos, mientras que la efectividad en la clasificación de personas, es prácticamente la mismas para ambas resoluciones.

Sobre los ajustes efectuados sobre el tamaño de resolución de la cámara, en donde se realizaría el análisis, se relaciona con la altura dispuesta de la misma. Ya que, mientras mayor sea la altura, mejor se puede apreciar el área de detección, y la resolución utilizada permite visualizar de manera clara el sector donde se efectuará la clasificación. Además, mientras más grande la

resolución, mayor es el retardo producido en la proyección del contenido. Por contraparte, mientras más pequeña la resolución, menor es el retardo, pero también disminuye la detección de personas.

**Respecto a la resolución:** la elección de la resolución con las que se efectuaron las pruebas tiene directa relación con el máximo provecho que puede obtenerse del sistema, esto ya que los 176x144 píxeles si bien tienen un menor tiempo de retardo, sus limitaciones al momento de la detección son muy importantes debido a que necesita de una silueta de un tamaño importante para identificar a una persona como tal, además que funciona de mejor manera situando la cámara en una posición no demasiado elevada. En cambio la resolución de 352x288, si bien posee un mayor tiempo de retardo en la obtención de la imagen, el tamaño de los píxeles utilizado por las personas a analizar es menor a lo que se requiere en la configuración anterior, además logra adaptarse de mejor manera a las distintas situaciones expuestas, ya sea por la altura en que se encuentre la cámara, como también el reconocer de forma más amplia el área de detección en la que se está trabajando, es por este motivo que se decidió utilizar este tipo de tamaño de video.

**Respecto al flujo controlado de personas:** el motivo de realizar este tipo de pruebas permitió definir, en primer lugar, la posición de la cámara en una altura definida, en donde se hizo posible obtener un mayor número de detecciones de manera correcta, esto debido a que el descriptor HOG logra separar las siluetas de las personas, a pesar de su proximidad de unas con otras, el contrario de los que sucede en una imagen de manera más frontal, en donde no se tiene noción de las profundidades, impidiendo separar a una persona de otra en caso de que transiten a distancias muy cercanas. Esto no implica que la detección a baja altura sea precaria ya que mientras exista separación entre las personas, funciona de manera correcta. Esto se puede observar en la prueba donde se colocó la cámara a 110 centímetros altura, en donde al considerar ciertas restricciones, principalmente la separación entre las personas, las detecciones fueron correctas.

**Respecto al flujo no controlado de personas:** los resultados son variados ya que existen diversos factores que afectan la detección, por ejemplo el lugar específico en donde se desea llevar un control estadístico de las personas es muy relevante, ya que una escena con muchos detalles afecta directamente con la obtención del histograma y, por lo tanto, la detección efectiva de las mismas. Además los resultados pueden variar según la iluminación presente en la escena, junto con las vestimentas utilizadas por las personas, ya que los colores también afectan al momento de realizar una obtención de HOG, ya que los colores pueden confundirse con los presentes en la escena.

Por lo tanto es necesario determinar el lugar en dónde se desee llevar un control en el flujo de personas, para así obtener el máximo provecho de las condiciones del mismo, para obtener el mayor número de detecciones de forma correcta, sin la necesidad de realizar modificaciones importantes en el sistema. Esto, debido a que su estructura tiene que ser tolerante a las distintas situaciones a las que pueda ser sometido. Sin embargo, no en todas las dependencias se cuanta con las mismas características físicas, como en donde se realizaron las pruebas, por lo que se está obligado a realizar algún tipo de ajuste, por ejemplo, en la disposición de la cámara.

**Respecto a la efectividad en detecciones:** con las distintas pruebas realizadas, se ha logrado determinar que el número de detecciones efectivas tiene directa relación con la configuración del programa que se está desarrollando, junto con las características del entorno en donde se realiza el análisis. En ese sentido, los experimentos realizados en un ambiente externo, son los más desfavorables, ya que se encuentra un gran número de elementos que afectan la obtención de las gradientes orientadas de la imagen al momento de analizar la imagen, por ejemplo árboles, postes, bancas, autos, entre otros. Por otra parte, los resultados obtenidos al interior de un edificio son mucho mejores, ya que es posible controlar en gran parte los elementos que pueden interferir de manera directa estas detecciones. En ese sentido es como se logró obtener más de 90% de efectividad, contra el 60% u 80% obtenidos en los experimentos realizados en el exterior.

**Respecto al sistema propuesto:** como se ha mencionado anteriormente se definieron parámetros y situaciones que permitieron obtener un gran número de aciertos en cuanto a la detección de personas, en ese sentido es necesario señalar los puntos claves para una correcta utilización del sistema propuesto. En cuanto a la configuración propuesta se presenta lo siguiente: La resolución de las imágenes utilizadas para el entrenamiento del detector es de 64x128 píxeles, resolución utilizada además, por Dalal y Triggs en su trabajo. La separación que se encuentra entre las clases y el hiperplano de SVM debe ser el mínimo posible, en ese sentido se recomienda que su valor sea 0.

En cuanto a la resolución de la imagen se ha optado por un tamaño de imagen de 352x288 píxeles, ya que esta configuración, junto con la disposición de la cámara a una altura de 3 metros aproximadamente, permite obtener los mejores resultados en cuanto a detecciones. Además el tiempo de retardo obtenido en cuanto a la captura de la imagen es de 3 segundos, siendo un tiempo aceptable para el sistema propuesto.

Es necesario señalar que a una altura demasiado elevada, el sistema no logra realizar detecciones de las personas, esto debido a que no es posible obtener una correcta imagen de las personas para su posterior clasificación, por lo que se descarta utilizar el detector en alturas superiores a los 5 metros.

## Discusión y conclusiones

En este trabajo de tesis se detalla el proceso realizado para la creación de un sistema que permite detectar personas, el cual fue implementado en una RPI 3. Para ello se utilizó el descriptor HOG, que corresponde al método con mejores resultados que se ha desarrollado en el último tiempo para efectuar este tipo de detecciones. Además se decidió trabajar con una RPI 3, ya que es uno de los computadores de placa reducida que cuenta con una relación precio/rendimiento con mejor evaluación en el mercado para proyectos relacionados con la electrónica.

En el primer capítulo se abordaron temas relacionados con el área del procesamiento de imágenes y visión por computador, con el objetivo de interiorizar al lector sobre las detecciones de objetos, principalmente, la detección de personas. Por lo tanto, es necesario conocer el cómo se realizan este tipo de procesamientos en las imágenes para obtener los resultados deseado, ya que existen distintas técnicas para realizar detecciones, como también para el reconocimiento. Además, se definió una problemática, consistente en cómo utilizar estas herramientas para la seguridad de recintos, como también el obtener estadísticas sobre el flujo de personas.

El capítulo dos, aborda los temas relacionados con la detección de personas mediante el descriptor HOG, para ello se analizó cada uno de los componentes que forman a este descriptor, desde la composición de un histograma en una imagen, hasta la obtención de los gradientes orientados. Además se expone el método utilizado para realizar la clasificación de una persona, la cual se realiza mediante SVM. El objetivo de esto es comprender el método empleado para la detección de personas, para en caso de ser requerido, realizar modificaciones en el programa desarrollado para obtener la mayor cantidad de aciertos en las detecciones efectuadas.

Es así como este capítulo, se basa principalmente en la obtención de gradientes orientados para la detección de personas, en donde para lograr esto se debe en primer lugar definir una imagen de entrada, la cual debe tener un tamaño de 128x64 píxeles, a esta imagen se le aplica una máscara (filtro) horizontal y vertical para obtener la dirección y magnitud de la gradiente. Es así como la imagen se descompone, posteriormente, en celdas de 8x8 píxeles, y luego en bloques de 2x2 celdas. Estos bloques se deslizan por toda la imagen, realizando un solapamiento entre ellos del 50%, lo que permite tolerar de mejor manera los cambios de luminosidad. De esta forma, cada bloque cuenta con 4 histogramas de gradientes, que se encuentran separado en 9 contenedores desde los 0° hasta los 180°.

El tercer capítulo aborda lo relacionado con la Raspberry Pi 3, sus características generales, los elementos que la componen y las mejoras que posee en comparación a sus antecesoras. Además se menciona que si bien, por defecto se puede programar en lenguaje Python, es posible trabajar con otros lenguajes, solo se necesita un editor de texto o un compilador que soporte el lenguaje deseado. En este caso se optó por C++, ya que se trata de un lenguaje mucho más directo en cuanto a orden y ejecución, haciendo que el programa tarde menos tiempo en realizar las tareas solicitadas. Esto producto de que las características técnicas de la RPI no se asemejan a la de un computador convencional, por lo que este lenguaje podría beneficiar el funcionamiento del programa.

Por otro lado, se presentan dos alternativas a la RPI 3, que corresponden a placas UDOO, las cuales poseen procesadores Intel, ya que en caso de no lograr implementar el sistema en la RPI, se deben analizar diferentes propuestas para su realización. La dificultad se encuentra en su alto valor comercial, por lo que se hace difícil su adquisición.

El cuarto capítulo consiste en la presentación del sistema desarrollado para la detección de personas, por lo tanto fue necesario detallar todo el proceso realizado para la obtención de un programa que cumpla con las expectativas de obtener una alta cantidad de detecciones efectivas, una baja tasa de falsos positivos y el mínimo tiempo de retardo entre la obtención de la imagen y su respectivo procesamiento.

Para la realización del detector, se utilizó el compilador Codeblocks y la librería OpenCV, esta librería cuenta con funciones que permiten realizar detección de personas mediante el descriptor HOG, por lo que cuenta con un clasificador mediante SVM, tal como lo presentado por Dalal y Triggs. De esta forma se presenta la estructura tanto de descriptor, como la estructura del sistema, para luego realizar pruebas que permitan obtener conclusiones importantes sobre la detección de personas, estas se realizaron tanto en computador como en la RPI, con el objetivo conocer el rendimiento de esta última, y así definir si es factible continuar con su utilización o no.

Es así como se procedió a analizar imágenes, para conocer el comportamiento del sistema, y al obtener resultados satisfactorios en ambos equipos, se decidió analizar videos. En ese aspecto, el rendimiento por parte de ambos computadores disminuyó considerablemente, pero en mayor grado afectó a la RPI, el motivo de este resultado tiene relación con las características del video y con las propias características de la RPI, impidiendo reproducir de buena manera el video y realizar detecciones al mismo tiempo. Sin embargo, la efectividad presentada en este caso es mucho mayor a los obtenidos en computador, el que además fue más rápido en lograr reproducir el video, pero afectando directamente a las detecciones. Esto se debe a que al tardar más tiempo en reproducir un frame, por parte de la RPI, el detector logra analizar de mejor manera la imagen, permitiendo efectuar buenas clasificaciones de las personas. Por este motivo es que las pruebas en tiempo real solo se realizaron en la RPI, ya que se determinó que es factible realizar este tipo de procesamiento en estas placas.

De esta manera se lograron obtener importantes conclusiones respecto al comportamiento del sistema, como por ejemplo el determinar una cantidad determinada de frame capturados por la cámara, en este caso se optó por 15 frame por segundo, además de observar que la resolución de

la imagen reproducida, tiene directa relación con la baja velocidad de reproducción del video, haciéndolo un sistema lento y con baja cantidad de detecciones. Por este motivo es que se decidió intervenir en ciertos parámetros ajustables, los que deberían mejorar la calidad y cantidad de detecciones, estos parámetros se analizaron en el siguiente capítulo.

El quinto y último capítulo presenta todos los ajustes realizados tanto en el programa desarrollado, como en los aspectos externos que pueden mejorar la detección de personas, como también así, los que afectan de manera directa al momento de realizar una clasificación para determinar si es persona o no.

Debido a esto, los primeros ajustes se realizaron sobre la resolución de cámara y la SVM, de aquí se puede concluir que a más pequeña resolución, menor es el retardo en la imagen obtenida, pero las detecciones también disminuyen, mientras que a las grande la resolución, las detecciones aumentan, pero con una retardo significativo en la reproducción del video, además se logra identificar un área determinada en donde se producen las detecciones. Por otro lado se definió la proximidad entre las clases y el hiperplano en la SVM, en donde mientras más cercano a él, menor es el riesgo de obtener un falso positivo en la imagen.

Otro aspecto que se estudio fue el de la altura de la cámara, en donde se comprobó que a mayor altura, mayor es la cantidad de detecciones efectuadas, esto es posible ya que se logra realizar una separación entre las personas, permitiendo detectarlas de manera individualizada (en la mayoría de los casos).

Los experimentos realizados en ambientes externos son muy complicados de analizar, ya que son variados los factores que influyen al momento de realizar una detección, por ejemplo los postes, las bancas, los árboles, etc. Por lo que mientras más plana sea la escena, mejores son las detecciones, debido a las pocas irregularidades presentes en el ambiente. Motivo por el cual se decidió realizar las pruebas de larga duración en el mismo lugar donde se realizó la configuración del sistema, ya que es un lugar muy transitado y cumple con las condiciones establecidas para obtener un buen número de detecciones.

Finalmente se puede concluir que el trabajo realizado cumple con el objetivo planteado al inicio del proyecto, ya que se logró implementar el sistema de detección de personas en una RPI, además se cumple con obtener una tasa de acierto de más del 90%. Sin embargo, se debe tener en consideración el ambiente en donde se realizan estas detecciones, ya que los factores externos, tanto como edificios, arboles, luz natural, reflejo en vidrios, entre otros, afecta de manera considerable la efectividad en las detecciones. Por lo que se recomienda implementar este tipo de sistemas, en interiores de edificios o lugares en donde el cambio de intensidad entre las personas y el entorno, puedan diferenciarse con mayor claridad, sin los excesivos detalles del terreno.

# Bibliografía

- [1] B. Escalante, «Procesamiento Digital de Imágenes» 2006.
- [2] «The History Of Image Processing Information Technology Essay». [Online]. Available: <https://www.uniassignment.com/essay-samples/information-technology/the-history-of-image-processing-information-technology-essay.php>. [Accedido: 10-Nov-2017].
- [3] D. Ballard and C. Brown, Ballard and Brown's Computer Vision.
- [4] R. Gonzalez and R. Woods, DIGITAL IMAGE PROCESSING, 2ND ED.
- [5] «RASPBERRY PI», 18-Dec-2013. [Online]. Available: <https://histinf.blogs.upv.es/2013/12/18/raspberry-pi/>. [Accesado: 7-May-2017].
- [6] «25 fun things to do with a Raspberry Pi». [Online]. Available: <https://www.cnet.com/how-to/25-fun-things-to-do-with-a-raspberry-pi/>. [Accesado: 22-Jun-2017].
- [7] «Computers in Spaceflight: The NASA Experience». [Online]. Available: <https://history.nasa.gov/computers/Ch9-3.html>. [Accesado: 8-Nov-2017].
- [8] J. Gómez, «Introducción al procesamiento digital de imágenes», 2001.
- [9] «Electromagnetic Radiation», 19-Sep-2015. [Online]. Available: [https://chem.libretexts.org/Textbook\\_Maps/Physical\\_and\\_Theoretical\\_Chemistry\\_Textbook\\_Maps/Supplemental\\_Modules\\_\(Physical\\_and\\_Theoretical\\_Chemistry\)/Spectroscopy/Fundamentals\\_of\\_Spectroscopy/Electromagnetic\\_Radiation](https://chem.libretexts.org/Textbook_Maps/Physical_and_Theoretical_Chemistry_Textbook_Maps/Supplemental_Modules_(Physical_and_Theoretical_Chemistry)/Spectroscopy/Fundamentals_of_Spectroscopy/Electromagnetic_Radiation). [Accesado: 19-Jun-2017]
- [10] «¿La visión del futuro?: Científicos a punto de crear ojos sintéticos con cámara y Wi-Fi», 21-Apr-2015. [Online]. Available: <https://actualidad.rt.com/ciencias/172672-vision-futuro-ojos-camara-wifi>. [Accesado: 21-Jul-2017].

- 
- [11] «¿Qué es la visión artificial?» [Online]. Available: <http://ingenierobeta.com/que-es-la-vision-artificial/>. [Accessed: 11-May-2017].
- [12] A. Estevenz, R. Verrastro, and L. Di Matteo, «Algoritmo de seguimiento de objetos basado en visión asistida por computador en tiempo real utilizando CAMShift e histogramas ponderados», Dec. 2014.
- [13] «Real Time Object Detection with TensorFlow Detection Model». [Online]. Available: <https://towardsdatascience.com/real-time-object-detection-with-tensorflow-detection-model-e7fd20421d5d>. [Accessed: 13-Nov-2017].
- [14] «OpenCV: How to Use Background Subtraction Methods». [Online]. Available: [https://docs.opencv.org/3.2.0/d1/dc5/tutorial\\_background\\_subtraction.html](https://docs.opencv.org/3.2.0/d1/dc5/tutorial_background_subtraction.html). [Accessed: 13-Jul-2017].
- [15] G. Welch and G. Bishop, «An Introduction to the Kalman Filter».
- [16] N. Dalal and B. Triggs, “Histograms of Oriented Gradients for Human Detection”, 2005, vol. 1, pp. 886–893.
- [17] R. García, «Detección y conteo de personas, a partir de mapas de profundidad cenitales capturados con cámaras TOF», 2015.
- [18] M. Pietikäinen, A. Hadid, G. Zhao, and T. Ahonen, Computer Vision Using Local Binary Patterns. Springer-Verlag London Limited, 2011.
- [19] «SIFT: Scale Invariant Feature Transform» [Online]. Available: <https://uilo.ubc.ca/sift-scale-invariant-feature-transform>. [Accessed: 13-Jun-2017].
- [20] H. Bay, A. Ess, T. Tuytelaars, and L. Van Gool, Computer Vision and Image Understanding, 2008.
- [21] G. Cheng Hann, «Local Feature Matching». [Online]. Available: <https://www.cc.gatech.edu/~hays/compvision/results/proj2/html/cgan30/index.html>. [Accessed: 2-Jun-2017].
- [22] «What is a Single-Board Computer (SBC)?» [Online]. Available: <https://www.techopedia.com/definition/9266/single-board-computer-sbc>. [Accessed: 26-Sep-2017].
- [23] «SBC, what is a single board computer, single board computers». [Online]. Available: <http://www.futureelectronics.com/en/display-solutions/single-board-computer.aspx>. [Accessed: 13-Sep-2017].
- [24] «About - OpenCV library». [Online]. Available: <https://opencv.org/about.html>. [Accessed: 12-May-2017].

- 
- [25] «OpenCV». [Online]. Available: <http://acodigo.blogspot.com/p/tutorial-opencv.html>. [Accessed: 15-May-2017].
- [26] C. McCormick, «HOG Person Detector Tutorial», 09-May-2013.
- [27] «Computer Vision: What is the difference between local descriptors and global descriptors?» [Online]. Available: <https://www.quora.com/Computer-Vision-What-is-the-difference-between-local-descriptors-and-global-descriptors>. [Accessed: 13-Jul-2017].
- [28] «Histograms». [Online]. Available: <https://statistics.laerd.com/statistical-guides/understanding-histograms.php>. [Accessed: 13-Jul-2017].
- [29] V. Atienza, «El Histograma de una imagen digital».
- [30] «Edge Detection». [Online]. Available: <https://www.mathworks.com/discovery/edge-detection.html>. [Accessed: 16-Jul-2017].
- [31] «Edge Detection». [Online]. Available: <http://fourier.eng.hmc.edu/e161/lectures/gradient/node6.html>. [Accessed: 16-Jul-2017].
- [32] S. Mallick, «Histogram of Oriented Gradients», Lear OpenCV, 06-Dec-2016. [Online]. Available: <https://www.learnopencv.com/histogram-of-oriented-gradients/>. [Accessed: 20-Jul-2017].
- [33] «Histogram of Oriented Gradients (and car logo recognition)». [Online]. Available: <https://gurus.pyimagesearch.com/lesson-sample-histogram-of-oriented-gradients-and-car-logo-recognition/>. [Accessed: 13-Jul-2017].
- [34] E. Carmona, «Tutorial sobre Máquina de Vectores de Soporte (SVM)», 11-Jul-2014.
- [35] «Raspberry Pi». [Online]. Available: <https://www.raspberrypi.es/que-es-raspberrypi.php>. [Accessed: 9-Jun-2017].

# A Detección de personas en RPI

## A.1 Código realizado

A continuación se presentará el código creado para la realización del detector de personas mediante el descriptor HOG en una RPI 3. Se recuerda que está escrito en lenguaje C++ junto con la utilización de la librería OpenCV.

Listado A-1: Código en C++ para detección de personas.

```
#include <opencv2/core/core.hpp>
#include <opencv2/highgui/highgui.hpp>
#include "opencv2/objdetect/objdetect.hpp"
#include <opencv2/imgproc.hpp>
#include <opencv2/video.hpp>
#include <iostream>

using namespace cv;
using namespace std;

int main( int argc, char** argv )
{
    VideoCapture cap(CV_CAP_ANY);
    if ( !cap.isOpened() )
    {
        cout << "No se puede reproducir video" << endl;
        return -1;
    }

    cap.set(CV_CAP_PROP_FOURCC,CV_FOURCC('Y','U','Y','V'));
    cap.set(CV_CAP_PROP_FRAME_WIDTH,352);
    cap.set(CV_CAP_PROP_FRAME_HEIGHT,288);

    int frame_width=cap.get(CV_CAP_PROP_FRAME_WIDTH);
    int frame_height=cap.get(CV_CAP_PROP_FRAME_HEIGHT);
    VideoWriter video("outD.avi",CV_FOURCC('M','J','P','G'),3,
    Size(frame_width,frame_height),true);

    while(true)
    {
        Mat img,img2,src;
        int counter = 0;
        if(counter % 15 == 0)
        {
            cap >> img;
        }
        else
        {
            counter++;
        }
    }
}
```

```

continue;
}

cvtColor(img, img2, COLOR_BGR2GRAY);
equalizeHist(img2, src);
HOGDescriptor hog;
vector<float> getPeopleDetector64x128();
hog.setSVMDetector(HOGDescriptor::getDefaultPeopleDetector());
vector<Rect> found, found_filtered;
hog.detectMultiScale(src, found, 0, Size(4,4), Size(0,0), 0.85, 2, false);

stringstream ss;
string ObjNum = ss.str();
ostringstream ContNum;

size_t i, j;
for (i=0; i<found.size(); i++)
{
Rect r = found[i];
for (j=0; j<found.size(); j++)
if (j!=i && (r & found[j])==r)
break;
if (j==found.size())
found_filtered.push_back(r);
}
for (i=0; i<found_filtered.size(); i++)
{
Rect r = found_filtered[i];
r.x += cvRound(r.width*0.1);
r.width = cvRound(r.width*0.8);
r.y += cvRound(r.height*0.07);
r.height = cvRound(r.height*0.8);
rectangle(img, r.tl(), r.br(), Scalar(0,255,0), 2);
}

for(int i=0; i<found.size(); i++)
{
ContNum <<i+1;
ObjNum = ContNum.str();
Point pCoordinates(found[i].x+10, found[i].y+25);
putText(img, ObjNum, pCoordinates, CV_FONT_HERSHEY_COMPLEX_SMALL, 1,
Scalar(0,255,0), 1, 8);
cout << "Cantidad de personas: " << ObjNum << endl;
ContNum.str("");
ContNum.clear();
}

imshow("Deteccion HOG", img);

video.write(img);

if(waitKey(30) == 27)
{
cout << "esc key is pressed by user" << endl;
break;
}
}
return 0;
}

```